NCS TIB 94-4

NATIONAL COMMUNICATIONS SYSTEM

TECHNICAL INFORMATION BULLETIN 94-4

VIDEO TELECONFERENCING

ADVANCED CODING TECHNIQUES

DECEMBER 1994

OFFICE OF THE MANAGER
NATIONAL COMMUNICATIONS
SYSTEM
701 SOUTH COURT HOUSE ROAD
ARLINGTON, VA 22204-2198

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1994	Final Report

4. TITLE AND SUBTITLE
VIDEO TELECONFERENCING ADVANCED CODING TECHNIQUES

5. FUNDING NUMBERS
C-DCA100-91-C-0031

6. AUTHOR(S)
Gary Rekstad

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Delta Information Systems, Inc.
300 Welsh Road, Suite 120
Horsham, PA 19044-2273

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Communications System
Office of Technology and Standards Division
701 South Court House Road
Arlington, Virginia 22204-2198

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NCS TIB 94-4

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The purpose of this study is to examine all advanced video coding techniques and determine which, if any, of the algorithms may yield significant enhancement to the ITU H.261 coding standard. The initial step in the study called for the preparation of an overview of advanced video coding techniques which is included. As part of this overview four advanced video coding systems were identified to be worthy of detailed study; vector quantization, wavelets, fractals, and object-based coding.

In order to obtain direct information on the performance of these techniques, commerical software packages for three coding systems [DCT/JPEG, wavelet, fractal] was obtained, and a comparative analysis was performed using typical teleconferencing scenes. Although this study is performed for intraframe coding the results are somewhat extensible to the interframe domain.

As one part of this project, a survey of commercially available advanced video compression systems was performed. The results of the survey are included. Finally overall conclusions and a bibliography are also provided.

14. SUBJECT TERMS
Video Coding Techniques
Video Teleconferencing
High Definition Television (HDTV)
Digital Facsimile

15. NUMBER OF PAGES
40 pages

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASS	UNCLASS	UNCLASS	UNLIMITED

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet *optical scanning requirements.*

Block 1. Agency Use Only *(Leave blank).*

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C - Contract	**PR** - Project
G - Grant	**TA** - Task
PE - Program Element	**WU** - Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

- **DOD** - See DoDD 5230.24, "Distribution Statements on Technical Documents."
- **DOE** - See authorities.
- **NASA** - See Handbook NHB 2200.2.
- **NTIS** - Leave blank.

Block 12b. Distribution Code.

- **DOD** - Leave blank.
- **DOE** - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
- **NASA** - Leave blank.
- **NTIS** - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only).*

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

NCS TECHNICAL INFORMATION BULLETIN 94-4

VIDEO TELECONFERENCING
ADVANCED CODING TECHNIQUES

DECEMBER 1994

PROJECT OFFICER

GARY REKSTAD
Electronics Engineer
Office of Technology
and Standards

APPROVED FOR PUBLICATION:

Dennis Bodson

DENNIS BODSON
Assistant Manager
Office of Technology
and Standards

FOREWORD

Among the responsibilities assigned to the Office of the Manager, National Communications System, is the management of the Federal Telecommunication Standards Program. Under this program, the NCS, with the assistance of the Federal Telecommunication Standards Committee identifies, develops, and coordinates proposed Federal Standards which either contribute to the interoperability of functionally similar Federal telecommunication systems or to the achievement of a compatible and efficient interface between computer and telecommunication systems. In developing and coordinating these standards, a considerable amount of effort is expended in initiating and pursuing joint standards development efforts with appropriate technical committees of the International Organization for Standardization, and the International Telecommunication Union - Telecommunication Standardization Sector. This Technical Information Bulletin presents an overview of an effort which is contributing to the development of compatible Federal, national, and international standards in the area of Video Teleconferencing. It has been prepared to inform interested Federal activities of the progress of these efforts. Any comments, inputs or statements of requirements which could assist in the advancement of this work are welcome and should be addressed to:

Office of the Manager
National Communications System
Attn: NT
701 S. Court House Road
Arlington, VA 22204-2198

TASK 3
TECHNICAL WORK IN THE
AREA OF VIDEO TELECONFERENCING

SUBTASK 2
ADVANCED CODING TECHNIQUES

FINAL REPORT
CONTRACT DCA100-91-C-0031
OPTION YEAR 3

December 13, 1994

Submitted to:
NATIONAL COMMUNICATIONS SYSTEM
ARLINGTON, VA

DELTA INFORMATION SYSTEMS, INC.
300 Welsh Road, Ste. 120
Horsham, PA 19044-2273
TEL: (215) 657-5270 FAX: (215) 657-5273

TABLE OF CONTENTS

APPENDICES

1.0 INTRODUCTION

This document summarizes work performed by Delta Information Systems, Inc. (Delta) for the National Communications System (NCS), Office of Technology and Standards. The NCS is responsible for the management of the Federal Telecommunications Standards Program, which develops telecommunications standards, whose use is mandatory for all Federal departments and agencies.

This document is a final report for a Task Order on Contract DCA100-91-C-0031. The titles for the contract and Task Order are listed below.

- Contract DCA100-91-C-0031
  Development of Federal Telecommunication Standards Relating to Digital Facsimile and Video Teleconferencing

- Task No. 3
  Technical Work in the Area of Video Teleconferencing

- Subtask No. 2
  Advanced Coding Techniques

The television world is experiencing a revolution caused by the conversion from the analog format to an all-digital structure. The revolution is universal including video telecommunications, broadcast TV [studio, tape recorders, direct broadcast satellites], HDTV, video teleconferencing, videophone, etc. This revolution has occurred as the result of one specific technology -- video coding which has achieved a high level of compression at reasonable cost. More specifically, virtually every digital system which is used on a widespread basis is based on one particular compression technique which has proven to be surprisingly robust and capable. This technique is based upon interframe prediction where the frame differences are encoded using the DCT transform in combination with motion compensation. The last public study of competing coding techniques was performed when the MPEG2 coding algorithm was selected based on subjective tests in Kurihama, Japan. Alternative coding techniques which were considered at that time included Vector Quantization and Sub-band coding.

The purpose of this study is to examine all advanced video coding techniques and determine which, if any, of the algorithms may yield significant enhancement to the ITU H.261 coding standard. The initial step in the study called for the preparation of an overview of advanced video coding techniques which is included in section 2.0. As part of this overview four advanced video coding systems were identified to be worthy of detailed study; vector quantization, wavelets, fractals, and object-based coding. Sections 3.0 through 6.0 of this report discuss these techniques of some detail, respectively.

In order to obtain direct information on the performance of these techniques, commercial software packages for three coding systems [DCT/JPEG, wavelet, fractal] was obtained, and a comparative analysis was performed using typical

teleconferencing scenes. The results of this study are included in Section 7.0. Although this study is performed for intraframe coding the results are somewhat extensible to the interframe domain.

As one part of this project, a survey of commercially available advanced video compression systems was performed. The results of the survey are included in section 8.0. Finally overall conclusions, and a bibliography, are provided in sections 9.0 and 10.0 respectively.

2.0 OVERVIEW OF ADVANCED VIDEO CODING TECHNIQUES

In order to fully understand *advanced* video coding techniques, it is important to first outline some of the basic principles of compression technology upon which the advanced concepts are based. First, the basic compression problem will be defined from a video conferencing perspective. Next, video compression will be viewed from a general perspective. Then, the basic ITU H.261 video coding algorithm will be described because it is this coding technique against which all *advanced* video coding techniques are compared. Finally, a brief overview of the advanced compression algorithms is provided.

2.1 The Compression Problem

The ITU H.261 video coding standard is universally used for virtually all video teleconferencing systems. The picture format most commonly used for videoconferencing is the Full CIF [Common Intermediate Format] which is defined below.

	Pixels/Line	Lines/Picture	Total Pixels
Luminance	352	288	101,376
Chroma R-Y	176	144	25,344
Chroma B-Y	176	144	25,344
			152,064

If the picture rate is 30 frames/sec, and each input pixel is encoded to 8-bit precision, the resultant bit rate of the uncompressed signal to be coded for transmission is 36.5 Mbit/s. Transmission bit rates for typical communication channels are 1.544 Mbit/s [a T1 channel], 384 Kbit/s, and 64 Kbit/s. The purpose of the video coder is to compress the input signal so that it can be transmitted through channels of this type. The required compression ratios for these transmission data rates are 24.3, 95, and 570 to 1, respectively. The compression problem is clearly a formidable one.

2.2 A Generic View of Video Compression

Figure 2.1 is a functional block diagram of a generic video compression system. It shows that the compressor is typically implemented in three sequential operations: signal analysis, quantization, and variable length coding. Basically, the signal analyzer performs measurements on the input uncompressed pixels. For example, the analyzer may compute prediction errors, compute transform coefficients, filter the signal into sub-bands, correlate the pixels with pre-stored VQ patterns, or correlate the pixels with the image itself (Fractals). The analyzer performs measurements within a single frame (intraframe) and/or from frame to frame (interframe). Typically, no compression is achieved by the signal analysis function. The input pixel data is merely transformed into another format which is more compressible than the original signal format. For example, in the DCT case , a block of 8x8 pixels is converted into an 8x8 array of DCT coefficients.

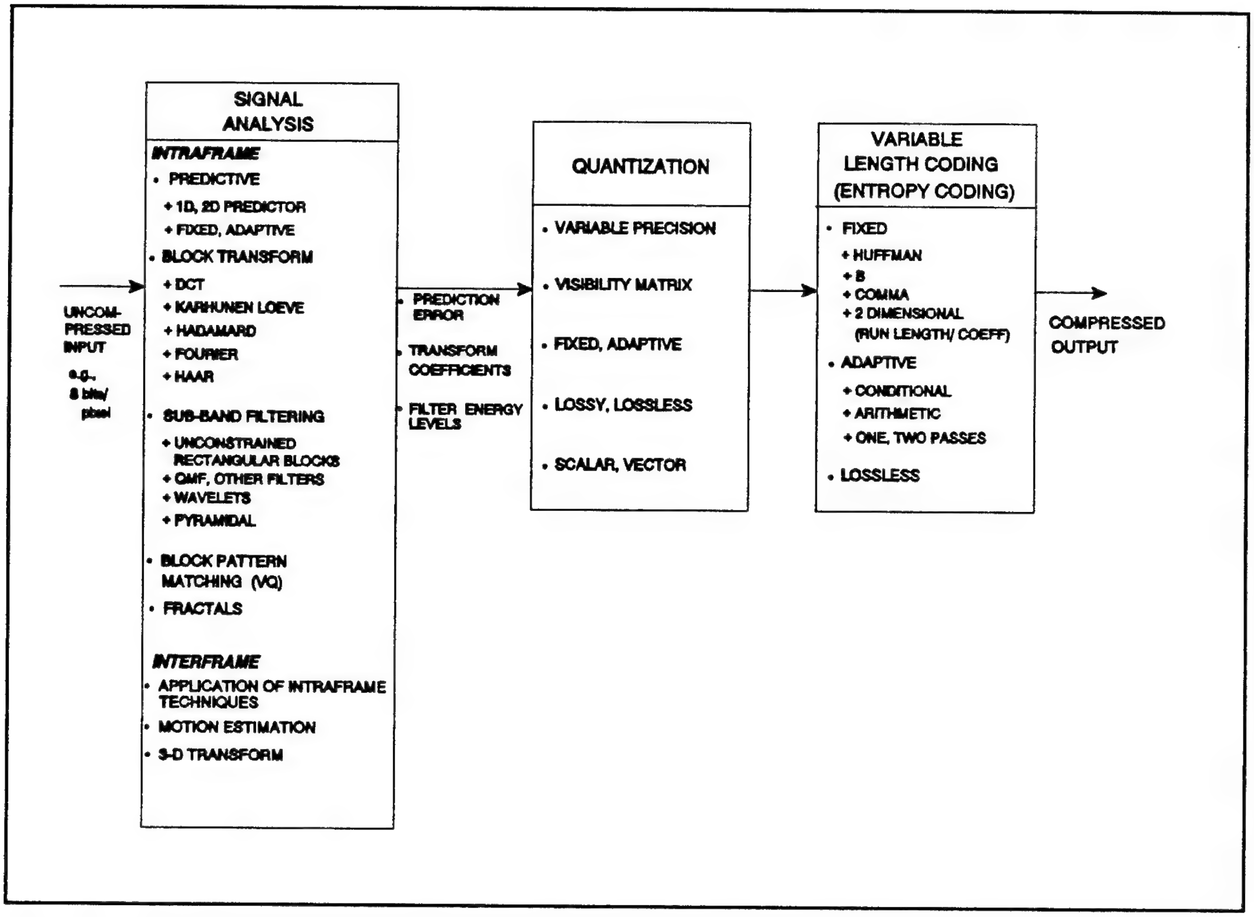


FIGURE 2.1 FUNCTIONAL BLOCK DIAGRAM - VIDEO COMPRESSION SYSTEM

The output of the typical signal analyzer process is usually quite accurate -- 8-bit to 12-bit precision. In lossless compressors, this accuracy is preserved in the quantization process. In lossy systems, the quantizer reduces the accuracy of the transformer output in a way that is as acceptable to the eye as possible. The quantization can be employed on either a scalar or vector basis. Most of the compression is achieved in this step by coarsely quantizing the transformed signal in such a way that any distortion to the eye is minimized. Typically, large numbers of coefficients are discarded because their value is low.

The final step in the compression process is to encode the quantizer output with a Variable Length Code (sometimes called Entropy coding). VLC is a technique whereby each event is assigned a code that may have a different number of bits. In order to obtain compression, short codes are assigned to frequently occurring events, and long codes are assigned to infrequent events. The expectation is that the average code length will be less that the fixed code that would otherwise be required. A major advantage of VLC is that it does not degrade the signal quality in any way (lossless). Therefore, the output picture quality is transparent to the VLC used.

2.3 The Standard Video Coding Technique [ITU H.261]

ITU Recommendation H.261 defines the video coding algorithm which is used in H.320 video conferencing terminals. Since this standard is used in virtually every video conferencing system, it is useful to briefly describe the coding technique so that it can be used as a reference for comparison to the advanced video coding techniques.

The H.261 coding algorithm is fundamentally based upon the Discrete Cosine Transform [DCT]. As shown in Figure 2.2, the picture is first segmented into blocks of 8x8 pixels. The DCT converts the 8x8 pixel array into an 8x8 array of DCT coefficients which are measures of the spatial frequency content of the block. For example, one coefficient measures the DC value of the block, and another is sensitive to a high frequency checkerboard pattern. Appendix 2A provides a more detailed description of the DCT and H.261 picture structure.

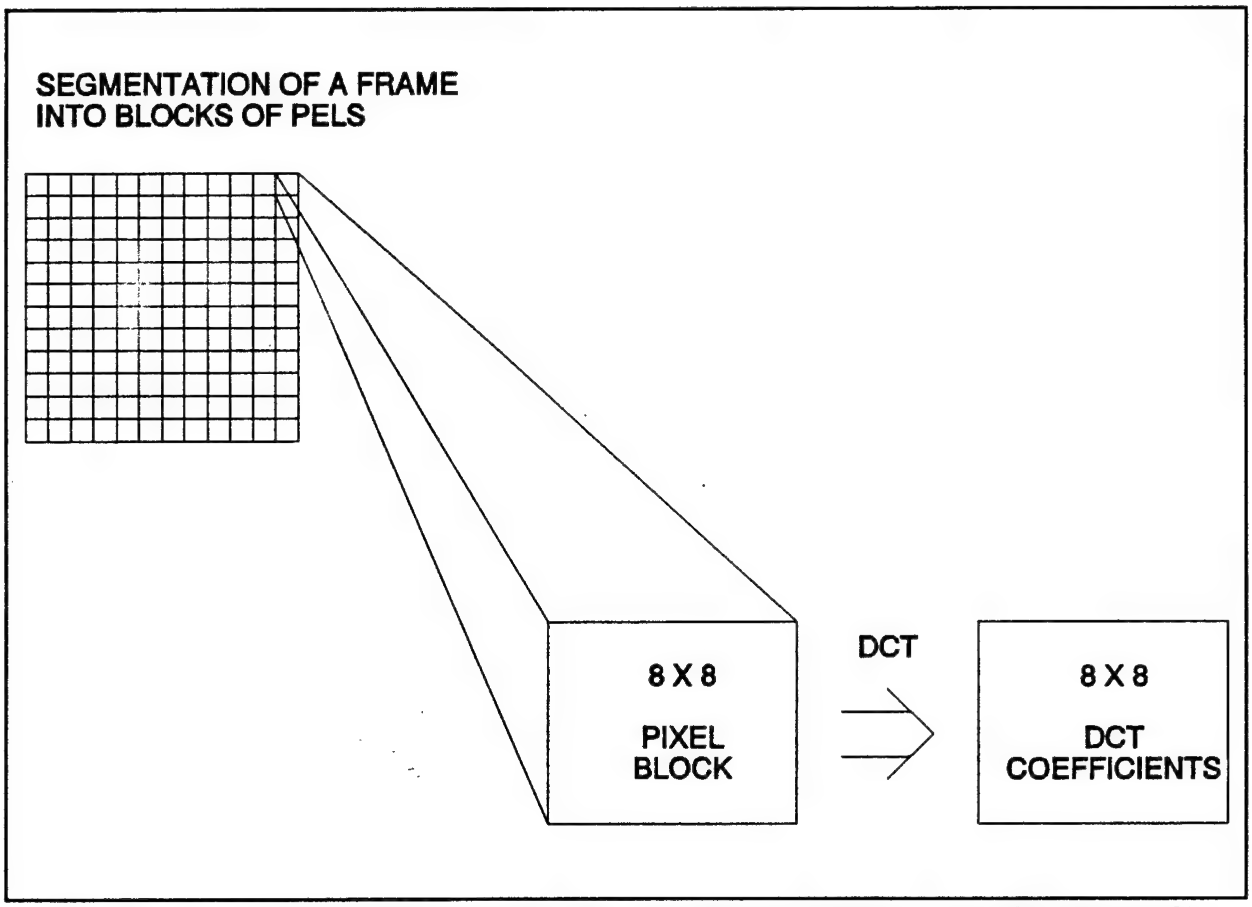


FIGURE 2.2 TRANSFORM CODING BY DISCRETE COSINE TRANSFORM (DCT)

Figure 2.3 is a functional block diagram of the H.261 video encoder. When the switch is in the INTRA position, the video is encoded by the DCT on an intraframe basis. However, the primary mode, which provides the greatest compression, is the interframe mode where the input video is compared block-by-block with a predicted picture, and the prediction error is DCT encoded for

transmission. Since adjacent pictures are very similar, this frame-to-frame compression is usually extremely high.

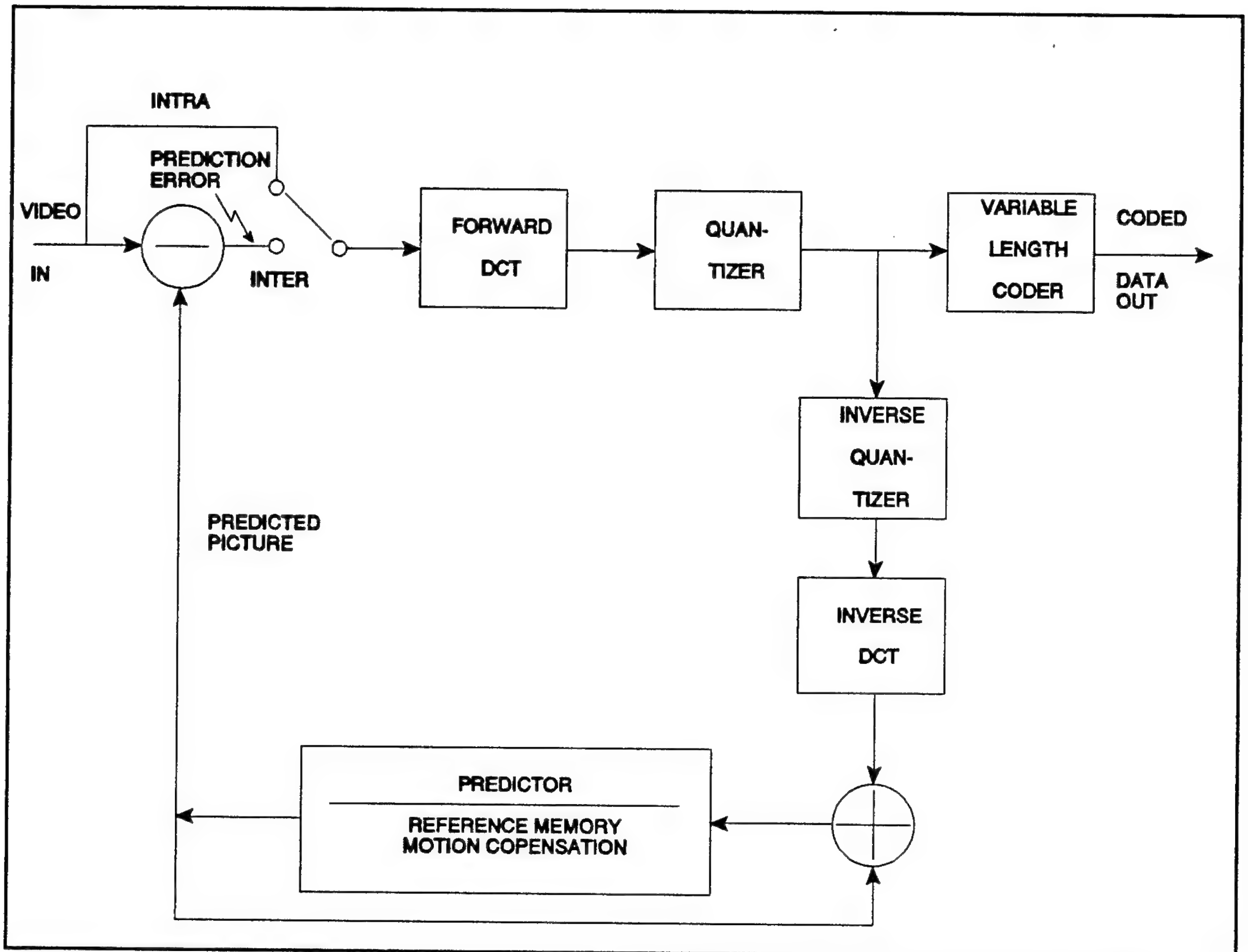


FIGURE 2.3 FUNCTIONAL BLOCK DIAGRAM OF THE H.261 ENCODER

The technique described above is extremely efficient when there is little, or no, motion in the scene, because the frame-to-frame prediction is very accurate. However, when motion occurs the prediction is no longer reliable without the use of an extremely important mode of H.261--Motion Compensation [MC]. At the encoder, MC measures the motion of the scene on a block-by-block [16x16 pixels] basis.

The operation of motion compensation is shown in Figure 2.4. Block "A" is a block in the current picture that is to be coded. Block "B" is the block at the same position as "A" but in the picture that was previously stored in both coder and decoder. Because of image motion, block "A" more closely resembles the pixel data from block "C" than that from block "B". The displacement of block "C" from block "B", measured in pixels in x and y directions, is the motion vector. The pixel-by-pixel difference between blocks "A" and "C" is transformed and coded. The motion vector and code data are transmitted to the decoder, where the inverse transformed block data is added to the data in block "C" pointed to by the motion vector, and placed in the block "A" position.

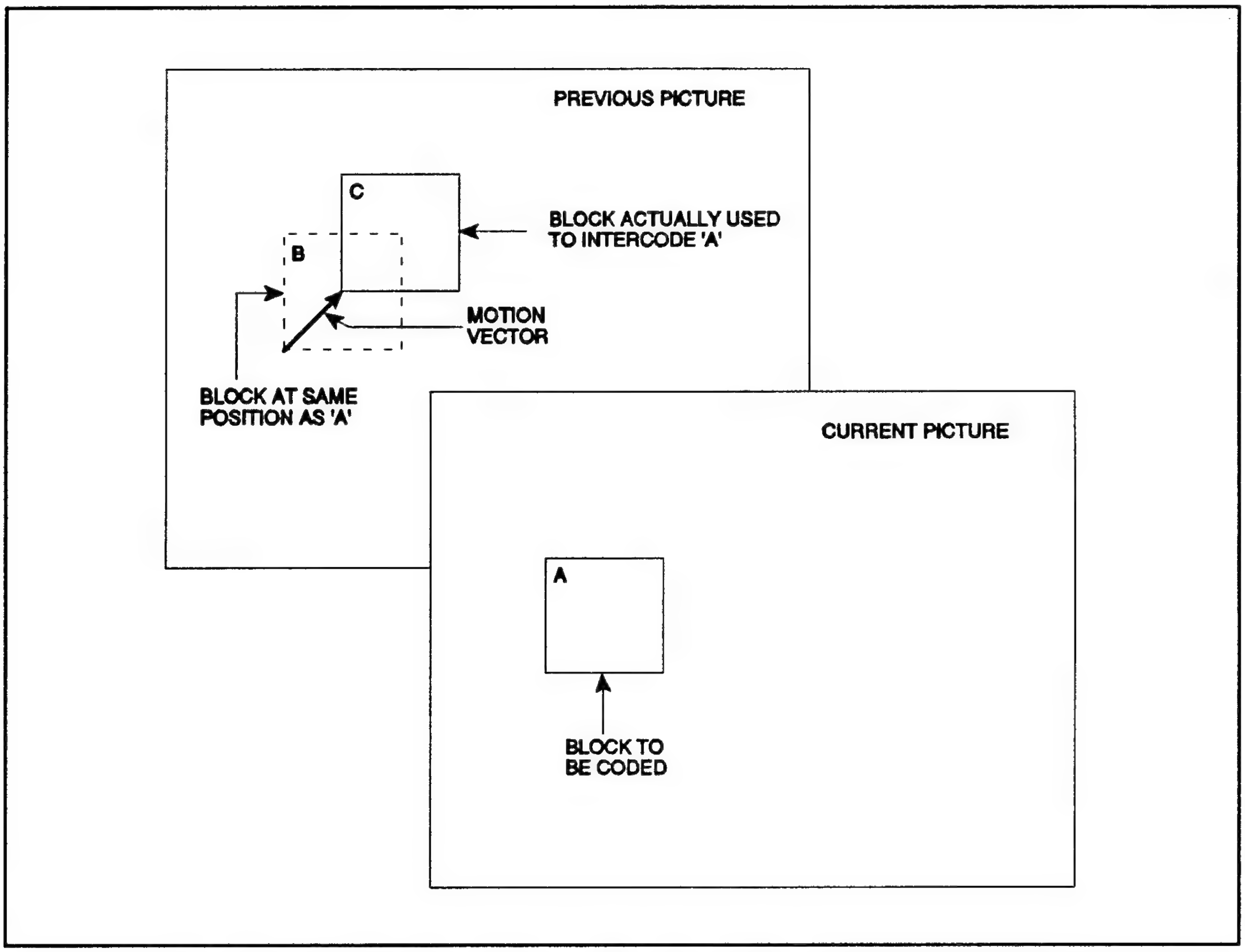


FIGURE 2.4 INTER-FRAME CODING WITH MOTION VECTORS

The ITU has begun work to develop an enhancement of the H.261 coding algorithm for very low bitrate applications (e.g. PSTN) such as the videophone and desktop videoconferencing. Typical enhancements which are being investigated are listed below.

- H.261 transmits motion vectors with a quantization precision of one pixel. It is clear that more precision is required for the videophone. Accuracies of one-quarter and one-half pixel are being studied. Independent studies are underway for both luminance and chroma motion compensation.

- H.261 employs motion compensation on a macroblock basis (16x16 pixels). Studies are underway to determine whether this also should be accomplished with more precision -- 8x8 or 16x8/8x16.

- VLC codes may be further optimized.

It is anticipated that these, and other possible improvement, will increase the performance by 1 to 3db.

2.4 Advanced Video Coding Techniques

The efficient coding of imagery and video signals has been the subject of research and development for many years, and the number of different compression algorithms which has been studied is extremely large. However, there are four coding techniques which experts generally consider to have the potential for possible improvement in performance over the H.261 standard. They are vector quantization, wavelet coding, fractals, and object-based coding. Table 2-1 summarizes two key characteristics of these techniques along with more basic coding techniques--PCM, DPCM, and DCT. Comments on each of these coding techniques is provided below.

TABLE 2-1

CODING TECHNIQUES	GEOMETRIC ELEMENT OF THE PICTURE BEING ENCODED	CONTENT BASED	SYMMETRY*
PCM	Pixel	No	S
Differential PCM	Pixel	No	S
Discrete Cosine Transform	Square Block of Pixels	No	M
Vector Quantization	Square Block of Pixels	No	MM
Wavelet	Multi-resolution Filtered Elements	No	M
Fractal	Block having any size or shape	Yes	MM
Object-Based Coding	Moving Objects	Yes	MM

* Complexity of the Encoder relative to the Decoder:
S - essentially the same;
M - more complex;
MM - much more complex.

PCM - Pulse Code Modulation is the basic uncompressed coding technique where each pixel is transmitted independently of its neighbors. Pixels are typically encoded with 8-bit accuracy to avoid visible distortion.

DPCM - Differential PCM is a fundamental compression algorithm which encodes the prediction error between the actual pixel value and a predicted value. Since it is fundamentally impossible to transmit less than one bit/pixel, the compression is limited.

DCT - The Discrete Cosine Transform transmits information about a block [typically 8x8] of pixels rather than individual pixels, and consequently is capable of much more compression as has been described in previous sections. The complexity of the basic DCT encoder is very similar to that of the decoder. However, the complexity of Motion Compensation in the encoder is quite high.

VECTOR QUANTIZATION - Like the DCT, Vector Quantization [VQ] transmits information about small square blocks of pixels. The size of a typical block is 4x4 pixels. Again like the DCT, high levels of compression can be achieved by analyzing the block on a macro basis. VQ has long been viewed as a potential improvement over DCT and H.261. However, VQ has gained a relatively high level of technical maturity and has not yet proven to be significantly superior to standardized techniques. The VQ encoder is much more complex than the decoder because of the extensive pattern matching search process which is required. Section 3.0 discusses VQ in more detail.

WAVELET CODING - Sub-band coding is a broad category of image coding which filters the input signal into sub-bands. By separating the video signal into different spectra, it becomes possible to process the different bands in ways that take advantage of the limitations of human vision. Wavelet coding is a particular type of sub-band coding which has received a lot of recent attention. It provides a high level of compression and is also reasonably simple to implement. Section 4.0 discusses Wavelet coding in more detail.

FRACTALS - Fractals have received a lot of attention recently for their possible application to image compression. In some ways fractal coding is similar to the DCT process. They both divide the input scene into contiguous blocks. The DCT uses square blocks of specified size, while fractal blocks can have any size and shape. They both use a type of "basis function" as a measure or correlation with an input block. In one case, the basis function is the DCT, in the other case the "basis function" is derived from the input image itself. For this reason, Fractal coding is considered to be content-based, while the process used in all previous techniques is considered to be independent of the content of the input scene. Fractals, like VQ, are highly asymmetric. The encoding process is extremely complex because the input scene must be analyzed to determine the best block boundaries and "basis functions". Section 5.0 discusses Fractal coding in more detail.

OBJECT-BASED CODING - Object-based coding [OBC] systems encode arbitrarily shaped objects moving within the scene instead of a set of contiguous blocks covering an entire picture. Consequently,the first step in OBC is the segmentation of a scene in moving objects and static background. Information is transmitted about the object defining its motion, shape, and color. The advantage of OBC is that the objects can potentially be displayed with better quality than block-based systems. The disadvantage is that the object segmentation process is complex and prone to error. Section 6.0 describes OBC in more detail.

3.0 VECTOR QUANTIZATION

Quantization is the process by which an analog signal, having a continuous range of possible values, is divided into a limited set of discrete steps. The most common quantization procedure is "scalar", where one can visualize a scale placed next to the variable being measured. In the scalar quantization process, one digital word represents the quantized value of one sample of a signal; for example, the value of a single pixel in an image.

In the case of Vector Quantization [VQ], one digital word represents the quantized value of <u>more than one</u> sample of a signal. In the case of the Vector Quantization of an image [References 11-19], a single word, or vector, is typically used to represent the quantized values of, for example, an entire 4x4 array of pixels. Unfortunately, it is difficult to visualize how one vector represents multiple quantized values in n-dimensional space. It is easier to visualize the vector quantization of an array of two adjacent pixels as illustrated in Figure 3.1. The figure shows the case where each pixel is first quantized, by conventional scalar means, to 4-bit [16 level] precision. The figure illustrates one vector representing the combination of brightness value 6 for pixel A, and brightness value 9 for pixel B. In general, the vector could take on 16x16=256 possible values; however, if only this procedure were employed, no compression would be achieved.

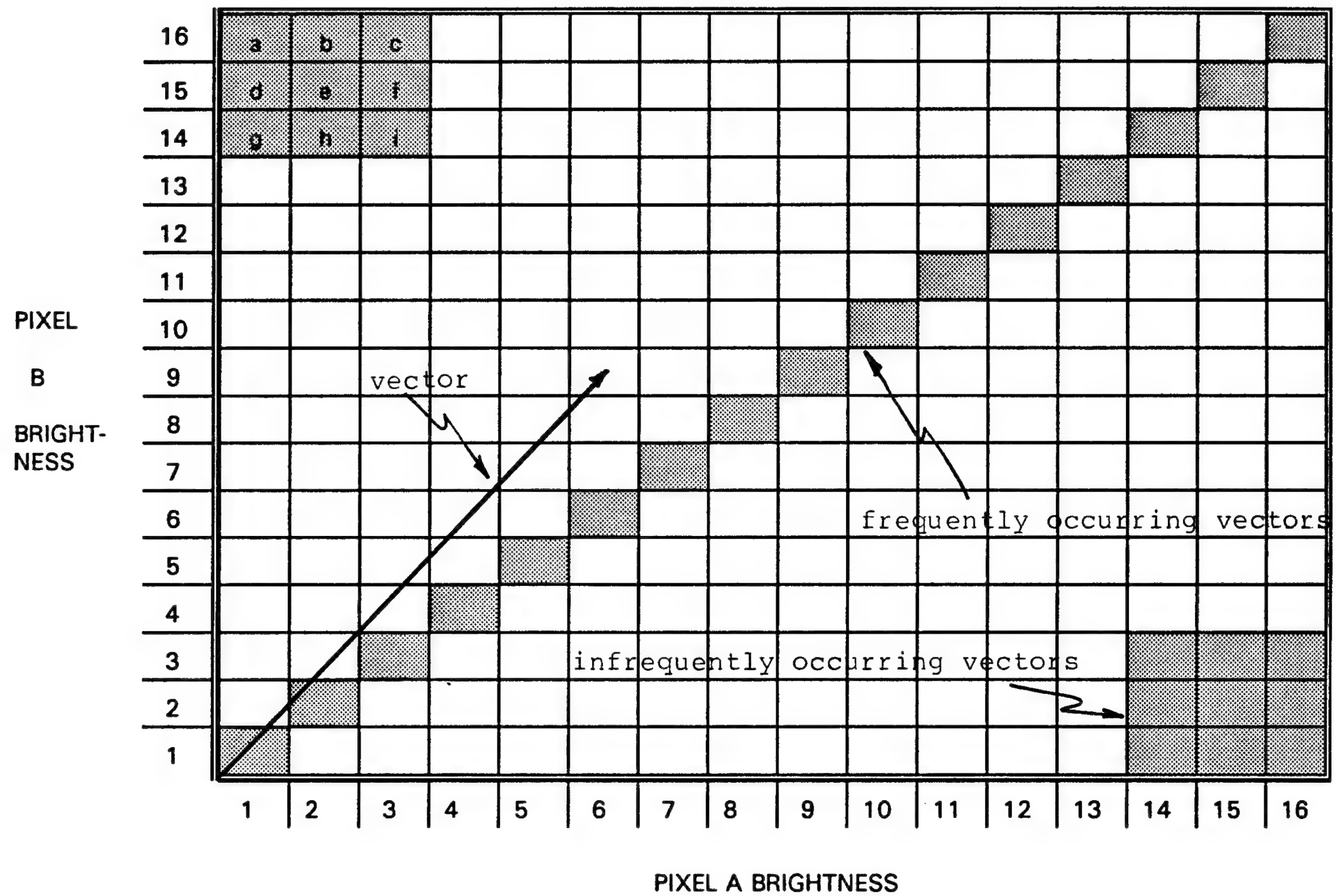
Compression in VQ systems is accomplished in two steps. In the first step, a number of adjacent vectors, which occur in a cluster in the input, are represented by a single vector at the center of the cluster. Clustering is accomplished using two approaches. In one case a very large sample of actual vectors from real images, usually known as a training vector set, is accumulated. From this set, vector clusters are located by rather complex algorithms, and representative vectors are placed at the centers of these clusters. The trade-off between distortion and compression is dependent on how many representative vectors are included in the list. The longer the list, the less the distortion, but also, the less the compression.

In the second approach clusters are defined on the basis of psychovisual perception. For example, the eye is very insensitive to variations in high contrast edges such as those occurring in the cluster of 9 vectors [a through i] in the corner of Figure 3.1. If any of these nine vectors [a-i] occur in the input image, one vector representing the cluster of vectors could be transmitted, and brightness values corresponding to vector "e" would be displayed at the output, without any perceived distortion. Little, if any, clustering is accomplished for equal-brightness vectors [45° diagonal in Figure 3.1] because they occur frequently, and the eye is sensitive to distortions in this region.

The lossy compression described above, merely sets the stage for the second step of compression which is lossless Variable Length Coding [also known as entropy coding]. This coding takes advantage of the fact that the vectors to be transmitted are not equally likely. Short codes are assigned to the most likely vectors, and long codes are assigned to the least likely. In this way the overall VQ

compression is a combination of psychovisual lossy compression and mathematical lossless compression. The above example illustrates VQ compression for a single image in two dimensions--x,y. Clearly VQ can be used even more effectively to compress a video signal in three dimensions--x,y,t.

One important characteristic of VQ is that it is highly asymmetric. That is, the encoder is far more complex than the decoder. The encoder requires a complex search process to make the decision as to what vector to transmit, while the decoder is merely a look-up table to display the value corresponding to the transmitted vector. This asymmetric characteristic make VQ very attractive for applications where video is encoded once [e.g. CD-ROM], and displayed many times on inexpensive display terminals.



SOURCES OF COMPRESSION

- NON-UNIFORM DISTRIBUTION (LOSSY)
- PSYCHOVISUAL (LOSSY)
- VARIABLE LENGTH CODING (LOSSLESS)

PIXELS

FIGURE 3.1 VECTOR QUANTIZATION

4.0 WAVELET CODING

The Wavelet transform [References 21-26] decomposes an image into frequency components. Unlike the DCT, however, the Wavelet retains the spatial context of the frequency components. This decomposition is performed by iterative low and high pass filtering using wavelet filters. The result of the high pass filter is retained. The result of the low pass filter is decimated and again low and high pass filtered. This is continued for the number of levels desired (typically 3 or 4). See Figure 4.1.

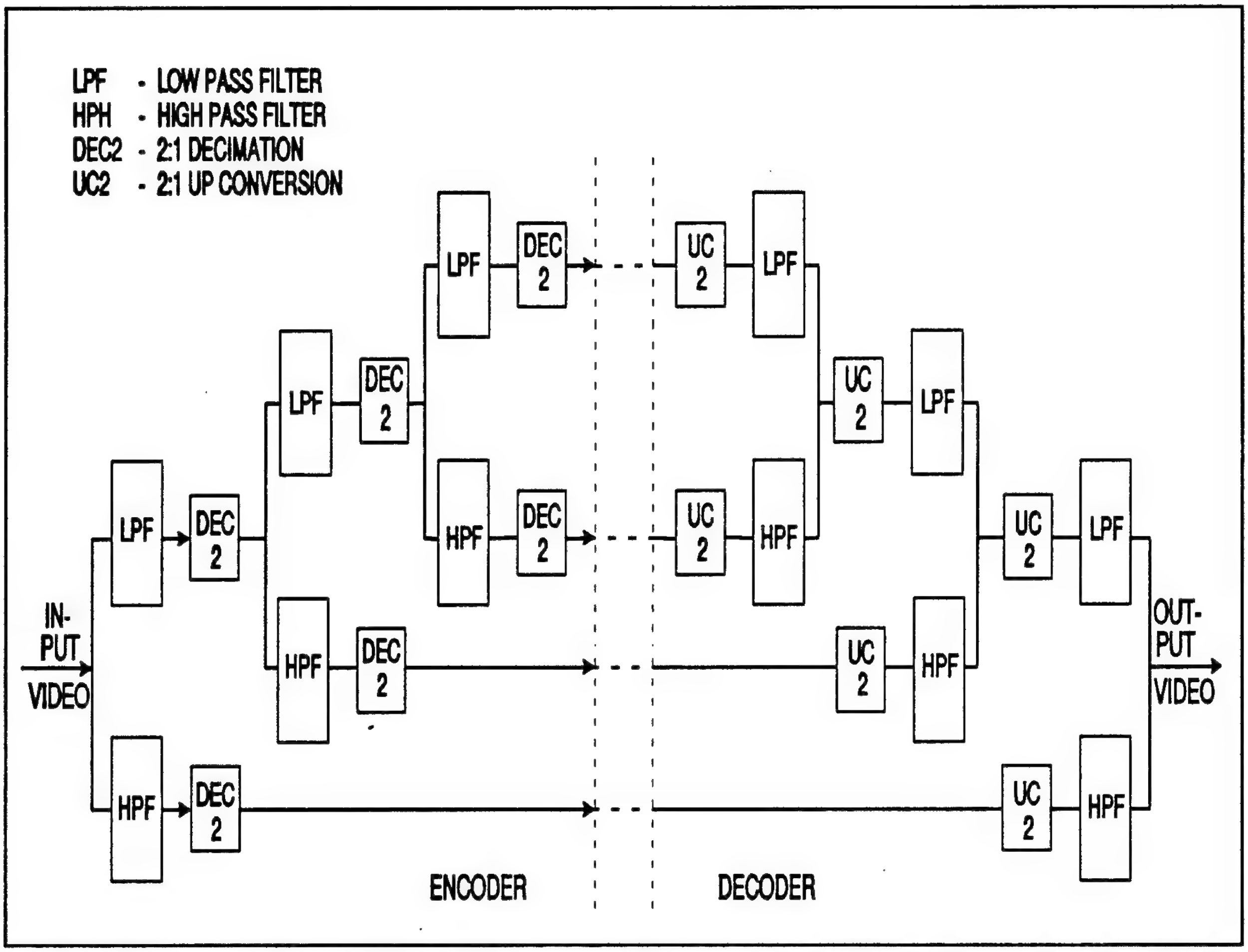


FIGURE 4.1 FUNCTIONAL BLOCK DIAGRAM OF A TYPICAL WAVELET CODEC

This filtering is performed on the entire image not on small blocks like the DCT. This eliminates the blocking artifact normally associated with DCT algorithms. At each frequency band, there is a filtered version of the image whose spatial resolution is proportional to the frequency. For example, low frequency bands have low spatial resolution that represent larger areas of the image. High frequency bands have high spatial resolution that represent local edge and texture detail. While the DCT coefficients represent a decomposition based on frequency related Cosines, the wavelet transform is a decomposition based on frequency related complex waveforms or wavelets. The result of the Wavelet transform is a hierarchical representation of the image where each level in the hierarchy is a map

of the image information contained within a specific frequency band. This hierarchy is represented by a tree structure of filtered pixels, with the lowest frequency pixels at the root. Each pixel at one level is linked to multiple pixels at the next level. The lowest frequency level is the called the reference signal. The other levels are the detail signals.

There are two important steps in designing a Wavelet compressor: 1) selection of the wavelet filter and 2) effectively quantizing the resulting tree structure. The wavelet filter is divided into two parts: an analysis filter and a synthesis filter. The analysis filter is used during compression and the synthesis filter is used during decompression. This filter pair must meet certain constraints. These are perfect reconstruction, finite length, and regularity. Perfect reconstruction indicates that a signal put into the input of the analysis filter is identical to the signal that comes out of the synthesis filter within some small error. Finite length means that it is implementable. Regularity means that the filter converges to a continuous function. Meeting these criteria, the filter is then selected to pack as much information about the original image as possible, into the reference signal. This is because the reference signal is generally quantized finer than the detail signals which are quantized coarsely or discarded.

The quantizer is the function where the compression gain is realized. The transform converts the pixels into a format that is more compressible. The quantizer does the compression. After quantizing the reference signal and detail signals, you are left with a sparsely populated tree of spatial and frequency position dependent information. This information is further compressed by effectively representing only the significant data without loosing the position information. This is done by tree trimming algorithms or run length coding. This leads to large compression ratios, while maintaining image quality and detail.

In decompression, the inverse quantized, filtered reference signal and detail signals are up sampled (interpolated) and applied to the synthesis filter banks. The outputs of these banks are summed. See figure 4-1. The result is the original image less any distortion due to quantization.

In Wavelet compression, the artifacts that occur at high compression ratios are loss of detail and ringing or oscillation around edges. The loss of detail is a result of the coarse quantization of the detail signals. The ringing is a result of the characteristics of the wavelet filters. Those filters exhibiting ringing in the impulse response of the filter will exhibit ringing to a greater extent in the resulting image.

There is work currently being done to adapt wavelet compression to motion video. There is not a single approach being used for interframe compression using wavelets. Several organizations both commercial and academic are investigating different techniques for interframe coding. One such technique takes the wavelet transform of the previous image and subtracts it from the wavelet transform of the current image. It then quantizes and transmits only the differences.

Another technique for motion video compression using Wavelets is Region Based Wavelet Transforms. This techniques identifies regions in the image. The regions are tracked for motion. Then wavelets are fitted to each region. Other techniques under study are Adaptive Wavelet Transforms, Overcomplete Wavelet Transforms, and Zero Crossing Translates. All of these techniques attempt to account for motion components (translations, rotations, scaling) in the wavelet transform.

5.0 FRACTAL CODING

A Fractal is a structure possessing similar looking forms of many different sizes. It has the properties of being infinitely magnifyable with structure at every scale and can be generated by small, finite sets of data and instructions. Fractal compression is based on three concepts: affine maps, iterated system functions, and the collage theorem. The affine map is a combination of rotations, scalings, and translations that act on a part of a source image to create a part of a target image. An iterated function system is a collection of contractive, affine maps. The iterated function system acts on a part of the source image to create a part of the target image out of repeated parts of the source image. These repeated parts are of various rotations, scales, and translations. The collage theorem says that if an image can be described by a set of affine maps then that set of maps provides an iterated system function which can be used to reproduce as good an approximation of the image as you desire.

The challenge is then to find a fractal model for a given image. This is done through the fractal transform. The fractal transform is a systematic method that breaks up an image into smaller regions, called domain regions, and finds the best affine maps for those regions. The domain regions are non-overlapping and completely cover the image. Range regions are also defined. They can overlap and do not need to cover the entire image. An affine map is generated that maps every range region into each domain region. The map and range region that provide the best match for a given domain region is selected. The affine coefficients, the domain region geometry, and the range region addresses for each domain range are packed to form the compressed data file, a FIF file, for still images or the compressed data stream for intra coded motion video. See Figure 5.1.

Decompression is performed by first creating two arbitrary buffers. The domain regions are identified on one buffer and the range regions on the other. The affine maps are applied using the range buffer and filling the domain buffer. The range and domain regions are then identified on the opposite buffers and the affine transforms are applied again. This is repeated until the differences between the two buffers are sufficiently small. This process results in an image which closely matches the original image before compression. See Figure 5.2.

There is work currently being done on motion video compression using Fractals. Images Incorporated offers a motion video compression product. It consists of a hardware card for compression and uses software for decompression. The product sells for about $12,000. The card plugs into the ISA bus in a PC, the software runs under windows.

Fractal coding of the difference images typically used in interframe coding does not work well because of the statistics of the difference image. One technique that is under study uses the difference image to identify areas where the prediction fails. The areas where prediction was satisfactory are defined as background, the areas where prediction failed are foreground. Those areas identified as background do not need to be transmitted since they are the same as

the previous image within a tolerable error. The foreground areas are then fractal coded and transmitted. Motion compensation techniques are not used.

FRACTAL TRANSFORM STILL IMAGE COMPRESSION

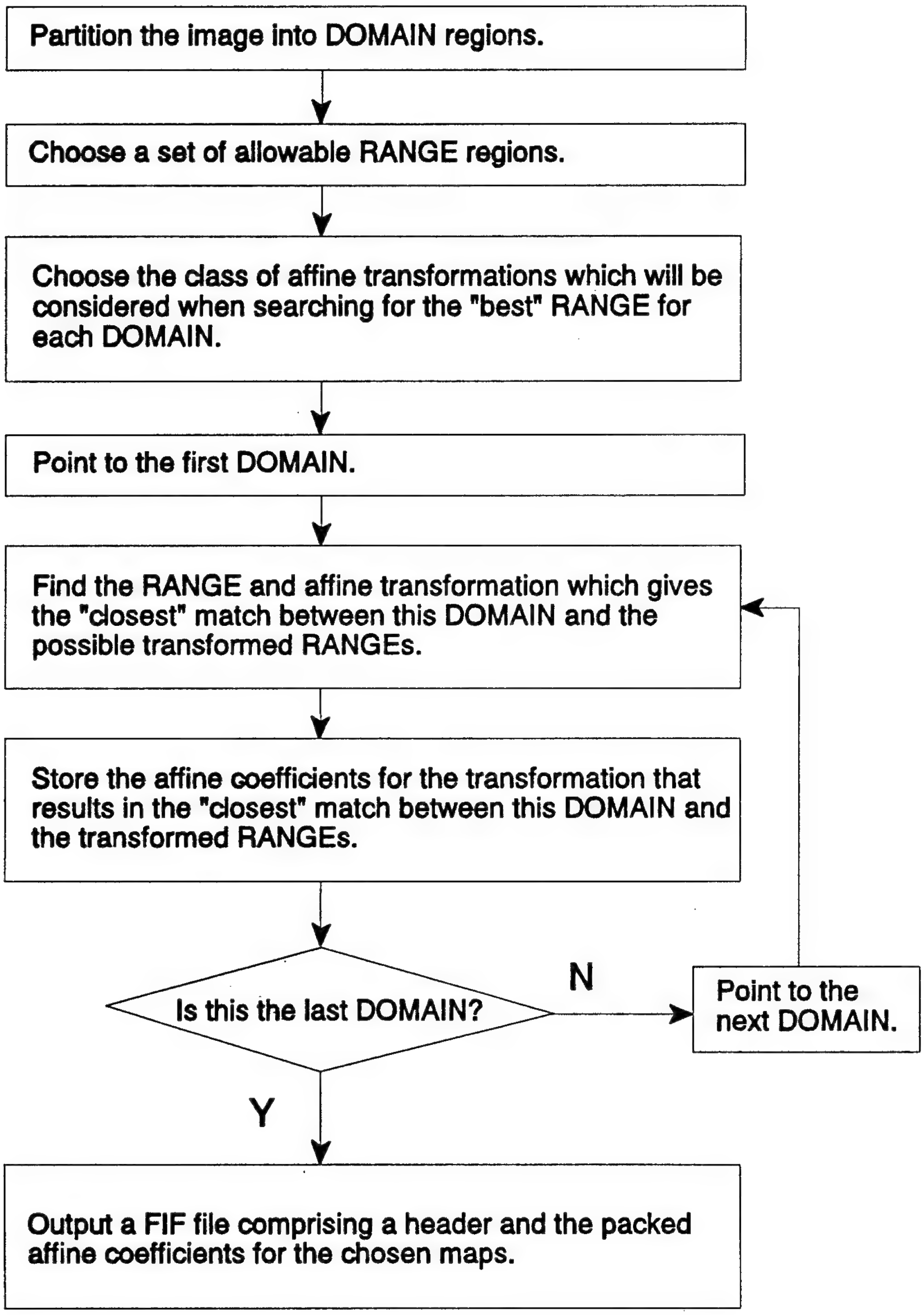


FIGURE 5.1

FRACTAL TRANSFORM STILL IMAGE DECOMPRESSION

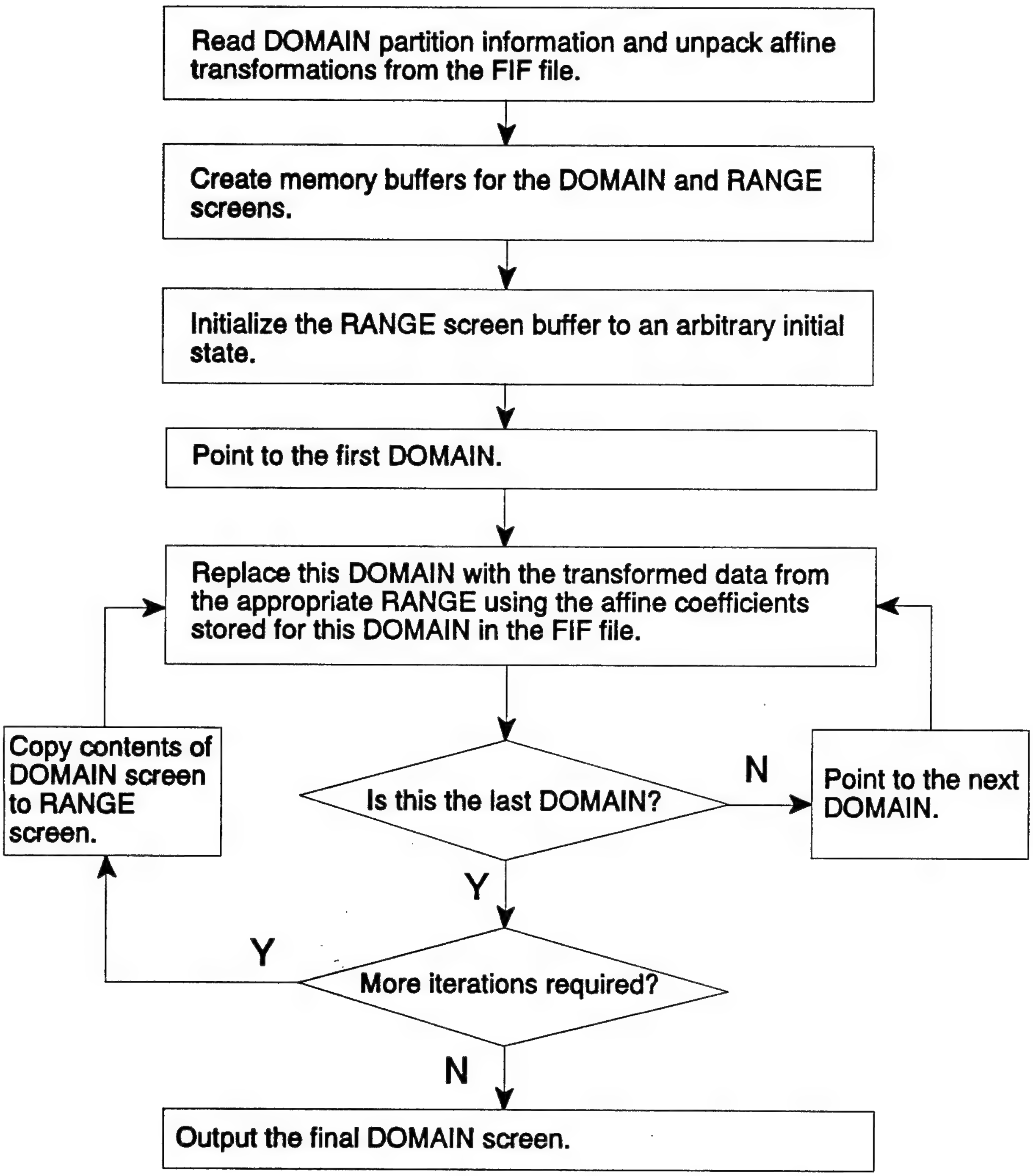


FIGURE 5.2

6.0 OBJECT-BASED CODING TECHNIQUES

One difficulty with the H.261 coding algorithm is that blocking artifacts and mosquito noise are frequently generated at the boundary between moving objects and the static background. This is caused by the fact that the DCT block edges are not aligned with the edge of the object. Object-based coding [OBC] has been devised to attempt to minimize distortions of this type. The general principle of OBC [References 36-43] involves the identification of, and the encoding of, arbitrarily shaped objects moving within the scene. Information is transmitted about the object defining its motion, shape, and color. Two types of object-based coding systems have been investigated--[1] a generic approach dealing with unknown objects, [2] those dealing with known objects such as a talking head [also known as knowledge-based coding]. The status of work in these two areas is outlined below.

6.1 Generic Unknown Objects

The University of Hanover, in Germany, has been a leader in the research of object-based coding. They have published many articles on the subject and have focused most of their energies on a specific implementation known as Object-Based Analysis-Synthesis Coding [OBASC]. Figure 6.1 is a functional block diagram of the OBASC encoder. The general architecture is the same as the H.261 coder in that they both employ interframe prediction and use the classic predictive loop structure. The Parameter Coder compares the Current Parameters with the Stored/Predicted Parameters and transmits the prediction errors to the receiver. The loop is a little more complex than normal because not only are current/predicted parameters being compared, but current/synthesized images are also being compared by the Image Analyzer. The Image Analyzer develops the current parameter set.

An example of image analysis for the test sequence "Miss America" is illustrated in Figure 6.2. The analyzer decomposes the input images of a sequence into differently moving objects. For each object, three sets of parameters are determined describing its shape, motion and color. In addition, each object is classified whether it complies with the underlying source model, i.e. whether the changes of the color parameters can be described only by that object motion which is allowed by the source model, or whether the source model fails. Figure 6.2 illustrates the processing of these two classes of objects--Model Compliant and Model Failure.

A great deal of research is being directed toward Source Models having a wide range of complexity. The simplest source model restricts the definition of the object to a rigid two-dimensional format. More advanced models are capable of defining the objects as having 2D/flexible, 3D/rigid, and 3D/flexible structures.

Research is also underway on a variation of object-based coding known as region-based coding. In general, a region is an area having uniform brightness. Consequently, an object is usually made up of a number of regions.

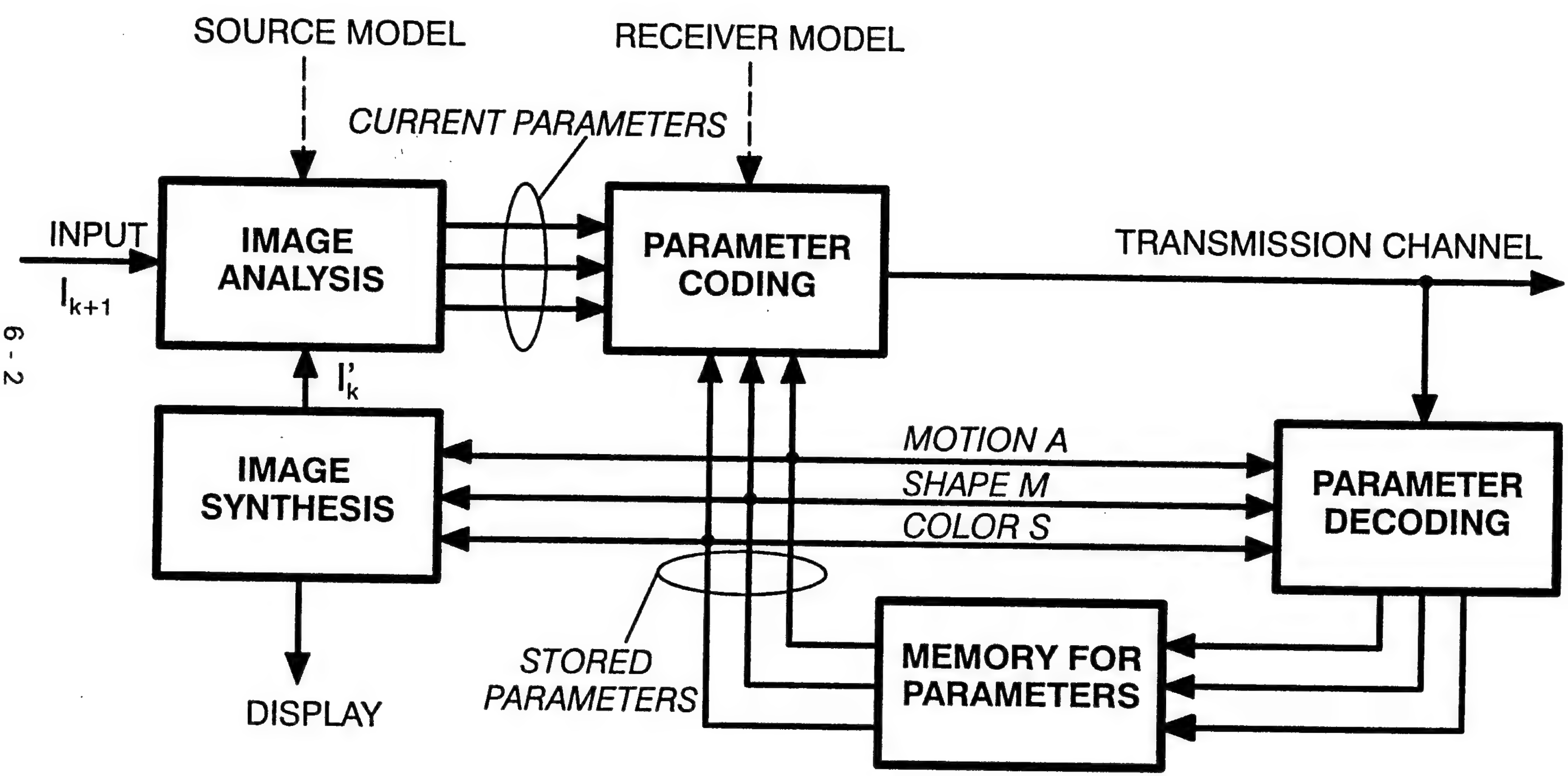


FIGURE 6.1 BLOCK DIAGRAM OF AN OBJECT-BASED ANALYSIS-SYNTHESIS CODER

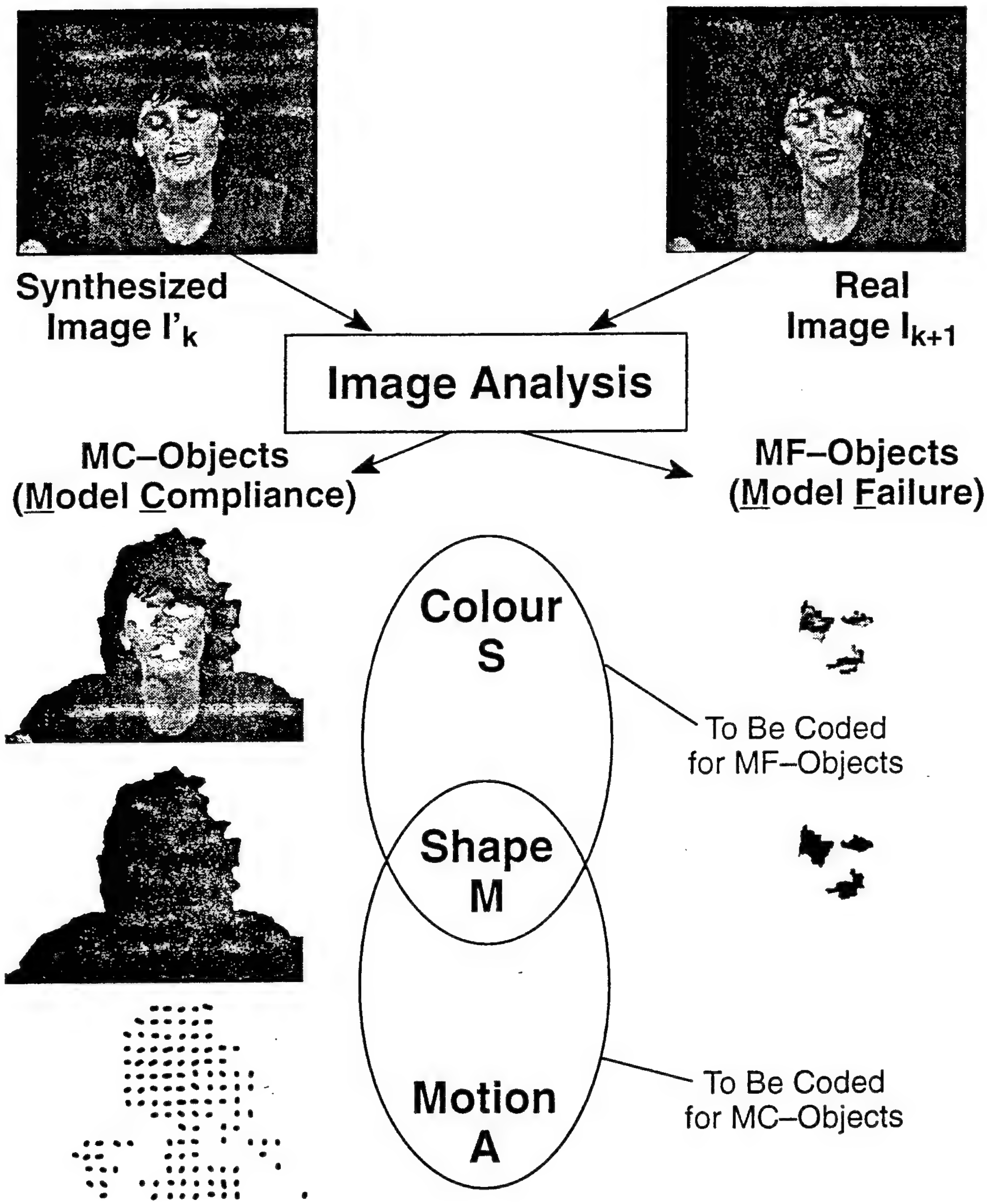


FIGURE 6.2 AN EXAMPLE OF IMAGE ANALYSIS OUTPUT
FOR THE TEST SEQUENCE "MISS AMERICA"

The advantage of region-based coding is that the requirements to define the texture of the region has been eliminated. The disadvantage is that the number of arbitrarily shaped elements to be defined and transmitted has increased.

6.2 Knowledge-Based Coding

The OBASC system described above is designed to detect, and encode, any generic objects which may appear in the scene. It does not take advantage of any apriori knowledge; e.g., it may be very likely that some objects [such as head and shoulders] will occur very frequently. Knowledge-based coding systems are specifically designed to take advantage of this prior knowledge. The knowledge-based object is usually defined by a wireframe model as illustrated in Figures 6.3 and 6.4. The first step in the encoding process in to detect the existence of the known object in the scene and the adaptation of the wireframe model to the characteristics of the particular object. At this point the normal OBASC procedures of transmitting motion, shape, and color parameters take place.

Research is underway on a more advanced version of Knowledge-based coding known as Semantic Coding. Semantic Coding is applicable to a knowledge-based object which has a restricted set of action units. In the case of the head-and-shoulders object, a restricted set of action units could be [1] mouth open/closed, [2] eye open/closed. An example of semantic coding for the Miss America test scene [reproduced from Reference 39] is provided in Figure 6.5.

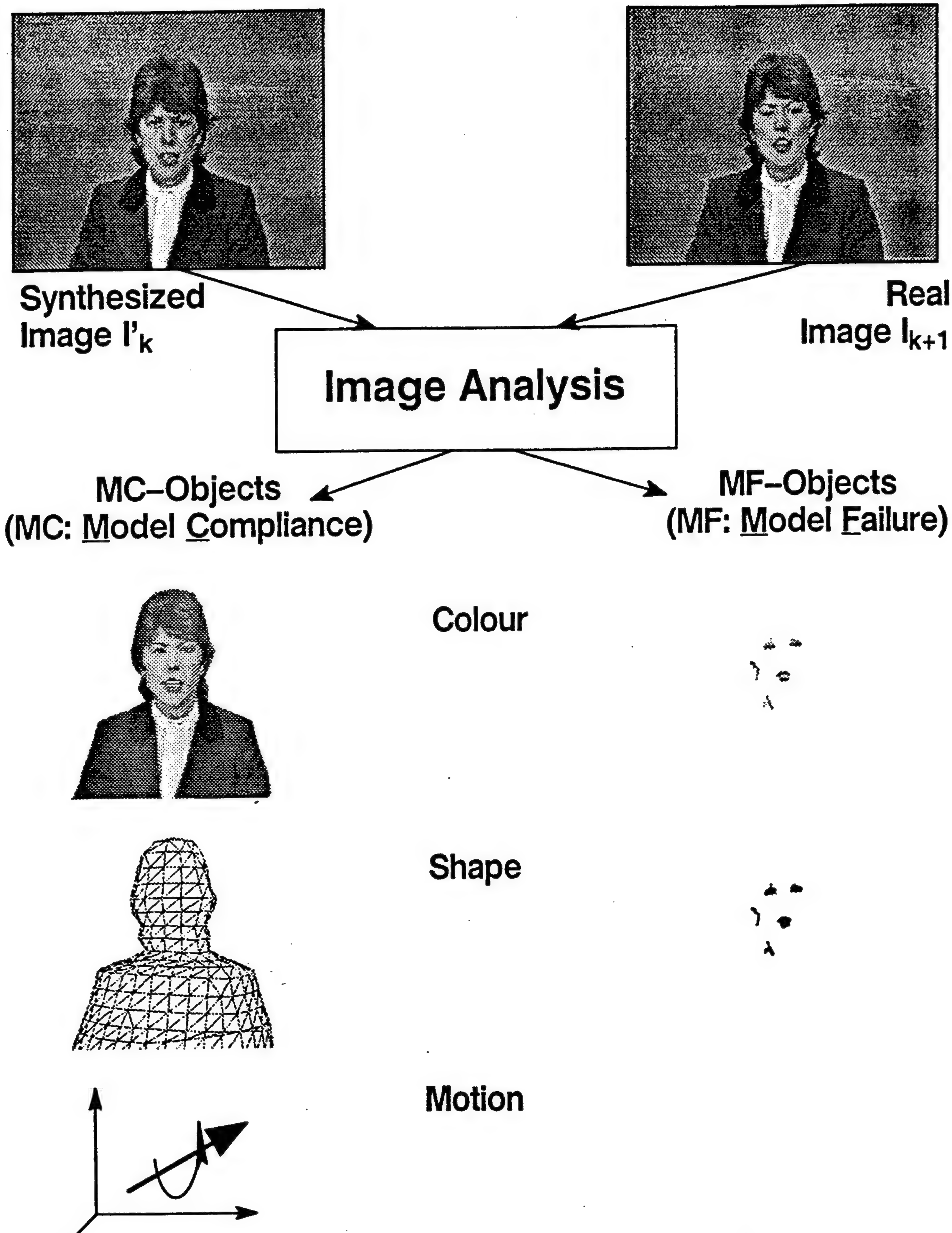


FIGURE 6.3 ILLUSTRATION OF MODEL-BASED IMAGE ANALYSIS

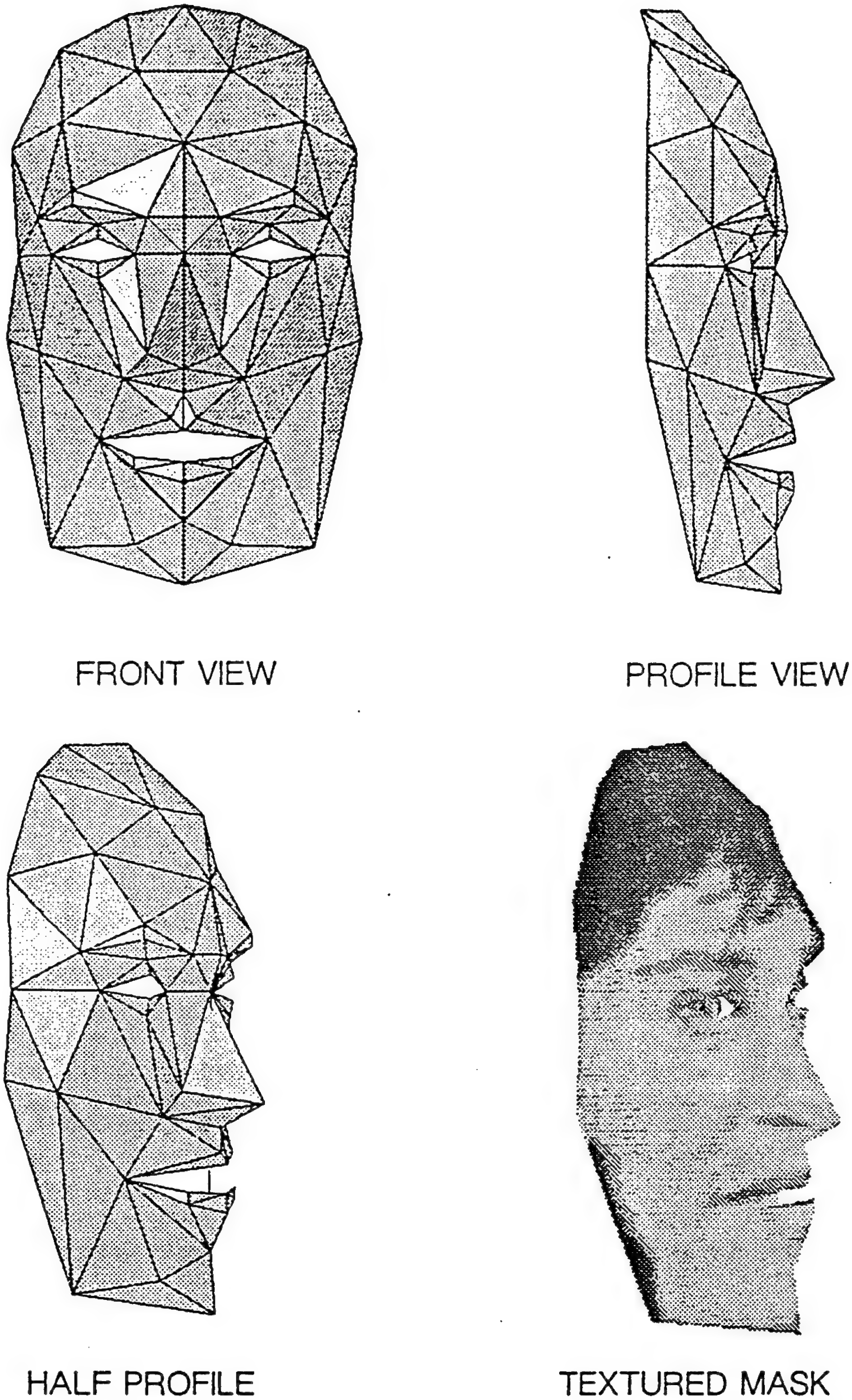


FIGURE 6.4 MODEL MASK CANDIDE

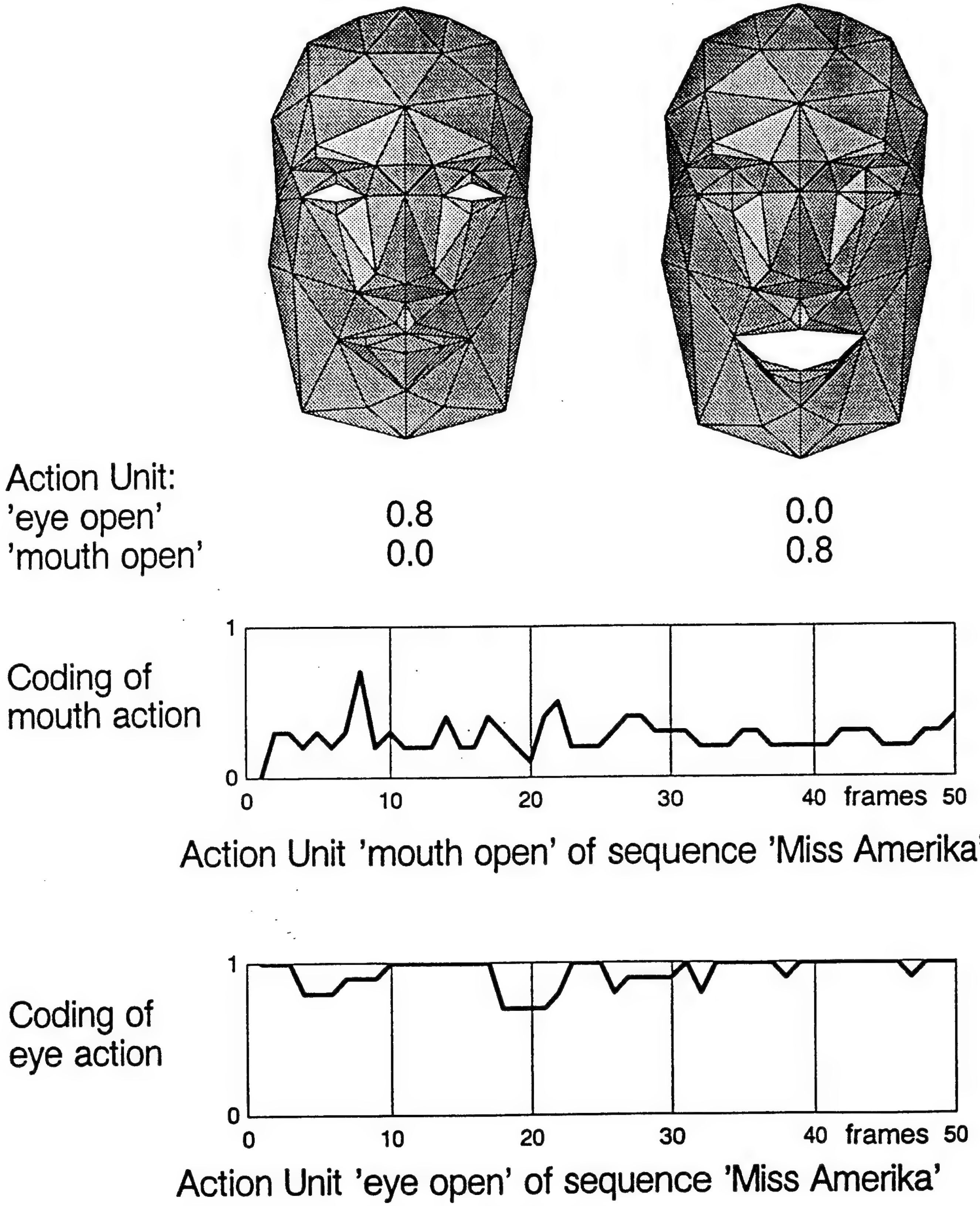


FIGURE 6.5 EXAMPLE OF SEMANTIC CODING

7.0 PERFORMANCE OF JPEG, FRACTAL, AND WAVELET PRODUCTS

7.1 Overview

In this task, Delta evaluated various advanced coding algorithms for image compression. The selected algorithms are based on Discrete Cosine Transforms (DCT), Fractal Transforms, and Wavelet Transforms.

The purpose of this task was initially to investigate motion video compression techniques that would be applicable to video teleconferencing, however, in the initial studies, we found that it was not possible to use these advanced algorithms for motion video. The reason for this was that the algorithms were not sufficiently mature for use with motion sequences. In the case of the Wavelet algorithm, work was only beginning to address motion video. In the case of the Fractal algorithm, compression for motion video could only be done using hardware. The cost of this hardware (> $10,000) was out of the scope of this project.

The task was modified to investigate the technologies using still images. This provides a valid investigation of the technologies since a key component of motion video compression is Intra-Frame compression. Intra-Frame compression is effectively the still image compression of a single frame without regard to the previous or next frame. A compression algorithm could code all frames in Intra mode (such as Motion JPEG), however, improved compression rates are realized by using Inter-Frame compression. In Inter-Frame compression, typically the current frame is subtracted from the previous frame and the difference is compressed. This is essentially the compression of a still difference image. Other techniques provide for not coding areas of the image that do not change much, or accounting for movement of objects between the two frames. These Inter-Frame techniques are applicable to all technologies being investigated. For this reason, a good measure of a technologies performance can be determined from the Intra-Frame performance.

7.2 Evaluation Description

The evaluation of the advanced algorithms will be performed through computer simulation. The primary goal of these simulations will be to determine performance and quality characteristics for each algorithm. These characteristics include the following:

- Compression Ratio
- Mean Square Error (MSE)
- Signal to Noise Ratio (SNR)
- Perceived (Subjective) Quality
- Coding Complexity

Compression ratio is determined by taking the ratio of the original image size to the compressed file size. For example, the VTC Group test image is 624 x 426 x

24 bits (797,472 Bytes). One of the JPEG compressed files, JGRP10.JPG, contains 77,642 Bytes resulting in a compression ratio of 10.3:1.

Mean Square Error (MSE) is calculated by taking the sum of the square of the difference between each pixel in the original image and the decompressed image divided by the number of pixels.

For T[i,j,k]=original test image, D[i,j,k]=decompressed image,i=vertical index, j=horizontal index, k=color index:

$$MSE = \frac{1}{(i * j * k)\ i, j, k} * \sum (T[i, j, k] - D[i, j, k]^2$$

Signal to Noise Ratio (SNR) is calculated by taking ten times the log of the ratio of the mean square of the original image to the mean square of the error image.

For T[i,j,k]=original image, D[i,j,k]=decompressed image,i=vertical index, j=horizontal index, k=color index:

$$SNR = 10\ LOG\left(\frac{\sum T[i, j, k]^2}{\sum (T[i, j, k] - D[i, j, k])^2}\right)$$

Subjective quality is determined by viewing the decompressed image next to the original image on the computer screen. Each decompressed image is given a score from the rating scale listed below:

SCORE	QUALITY
5	IMPERCEPTIBLE
4	PERCEPTIBLE BUT NOT ANNOYING
3	SLIGHTLY ANNOYING
2	ANNOYING
1	VERY ANNOYING

Complexity is measured by the amount of time it takes to compress the test images. Decompression time was not used, because some of the compression programs automatically displayed the decompressed image while others did not. This resulted in distorted measures of the decompression time.

The evaluations will be performed using test sequences characteristic of video teleconferencing applications. Three test scenes were selected. They are titled: VTC Group (Plate 7.1), VTC Individual (Plate 7.2), and Zelda (Plate 7.3). All three images are in 24 bit RGB format. The two VTC images are 624 x 426 and Zelda is 720 x 526.

The evaluation process consists of the following steps:

1) Apply compression algorithm to test image to produce compressed file.
2) Determine compression complexity.
3) Measure compression ratio.
4) Apply decompression algorithm to coded data to produce decompressed image.
5) Compare test image to decompressed image and compute MSE and SNR for quantitative evaluation of compression.
6) Display test image and decompressed image for subjective evaluation of compression.

In order to perform the simulations in a cost effective yet accurate manner, the simulations were performed on a high performance 66 Mhz 486 based PC system. The system has a high resolution 1280 x 1024 24-bit color SVGA monitor.

7.3 Product Descriptions

Delta has obtained copies of the compression algorithms which implement the desired technologies. These algorithms are described below.

7.3.1 JPEG

JPEG compression is based on the Discrete Cosine Transform (DCT). The DCT is a pixel transform that converts from the spacial domain to the spectral domain. This conversion compresses the majority of the image energy into a few of the spectral bands. This allows the other bands to be discarded without significant loss of quality.

The compression processes divides the image into 8x8 pixel blocks. The DCT is applied to each block. The resulting coefficients are then quantized and variable length coded. The majority of the compression is provided by the quantization step which essentially throws out information in the DCT coefficients. If this quantization is done effectively, the resulting image can be a very good representation of the original. In order to improve the image quality, different DCT coefficients are quantized differently. This allows the high frequency coefficients to be quantized more heavily than the low frequency coefficients. Since the Human Visual System is less sensitive to high frequency variations, this loss of high frequency information is less noticeable and produces higher quality images than a constant quantizer would.

The JPEG software performs a conversion from the 24 bit RGB 4:4:4 color space to YCbCr 4:1:1 format. Compression is performed on the YCbCr data. The decompression software converts the format back to RGB at 4:4:4.

The techniques used in JPEG are very similar to those used in H.261, the video coding algorithm used for video teleconferencing systems. The differences being that H.261 uses a constant quantizer for all DCT coefficients, and H.261 provided for Interframe compression.

The JPEG software used for this evaluation is from the Independent JPEG Group. It provides many controllable parameters, three are related to quality and compression ratio. The Quality parameter controls the scaling on the quantizer matrix. This effectively controls the quantization of the DCT coefficients. It varies from 0 (lowest quality) to 100 (highest quality) and was used to control the compressed file size. That is, lower quality produced smaller compressed file sizes, higher quality produced larger compressed file sizes. The Optimize and Qtable parameters were not used in this evaluation. These parameters allow generation of custom quantization tables and Huffman codes for the image. The default quantization tables and Huffman codes were used, because it would have been a major effort, outside the scope of this project, to determine better ones.

7.3.2 Fractal

Fractal compression is described in Section 5.0.

The fractal compression software used for this evaluation is Images Incorporated for Windows from Iterated Systems, Inc. This program provides software compression and decompression on still images. It also provides software decompression of motion sequences, however, a hardware compressor is required to compress motion sequences.

The Images Incorporated software performs a conversion from the 24 bit RGB 4:4:4 color space to an internal 8 bit format. Compression is performed on the 8 bit data. The decompression software converts the format back to RGB at 4:4:4.

The software has four compression parameters: Quality, approximate file size, archive, and template. Quality has four selections ranging from fair to best. Best was always used. Approximate file size was used to control the compression ratio. Archive and template were not used. Template requires another image file containing features of the image to be compressed. This file must be present for both compression and decompression.

7.3.3 Wavelet

Wavelet compression is described in Section 4.0.

The Wavelet compression software selected for this evaluation is SWIFTimage from Summus LTD. This package is a based on a proprietary wavelet transform. The transformed signals are then scalar quantized. The resulting quantized data is run length coded to remove the insignificant data and then entropy coded.

The SWIFTimage software performs a conversion from the 24 bit RGB 4:4:4 color space to YIQ 4:1:1 color space. Compression is performed on this subsampled YIQ. The decompression software converts the format back to 24 bit RGB at 4:4:4.

The software has several parameters for optimizing compression for various types of images and functions (for example: sparse images, focus, detail retention, magnification). The default was used for all of these parameters and a threshold parameter was used to control the compression ratio. This parameter controls the quantizer.

7.4 Results

The results of the objective and subjective evaluation for the test images and compression technologies are described in the following three tables. Table 7-1 shows the results for the JPEG compression algorithm. Table 7-2 shows the results for the Fractal compression algorithm. Table 7-3 shows the results for the Wavelet compression algorithm. The first column indicates the test image used in the evaluation. The second column shows the original test image size in bytes. The original image was 24 bits per pixel. The third column shows the compressed file size. This is the actual size of the compressed file in bytes. It contains all of the information required to decompress the file as well as any header information required by the algorithms. The fourth column is the actual compression ratio determined by dividing the original test image size (column 2) by the actual compressed file size (column 3). The fifth column is the compression time. This is an indication of the complexity of the compression algorithm and its applicability to real time use. However, it should be noted that complexity does not necessarily translate directly to system cost because inexpensive chips may be available (e.g. H.320) to implement complex functions. The sixth column shows the mean square error and the seventh column shows the signal to noise ratio. Both are objective measures of image quality. The eighth column lists the mean opinion score. Four observers rated the decompressed images along side the original image and scored the quality of the decompressed image according to the scale in section 7.2. The scores of the four observers were averaged and reported in the table.

TABLE 7-1
JPEG COMPRESSION RESULTS

TEST IMAGE	INITIAL IMAGE SIZE (BYTES)	COMPRESSED FILE SIZE (BYTES)	ACTUAL COMPRESSION RATIO	COMPRESSION TIME (SEC.)	MSE	SNR (db)	SUBJECTIVE (MOS)
GROUP	797,472 (624 x 426)	77,642	10.3:1	18	5.96	30.93	5
		32,132	24.8:1	18	11.23	28.18	4.75
		8,269	96.4:1	18	80.02	19.66	3.25
		4,795	166.3:1	18	348.37	13.27	1
INDIVIDUAL	797,472 (624 x 426)	78,417	10.2:1	18	14.43	25.80	5
		31,954	25:1	18	33.99	22.07	5
		7,974	100:1	18	258.62	13.26	2.25
		4,646	171.6:1	18	952.54	7.60	1
ZELDA	1,244,160 (720 x 576)	127,195	9.8:1	43	26.50	25.94	5
		50,367	24.7:1	42	29.39	25.49	5
		12,492	99.6:1	39	71.96	21.60	4
		7,309	170.2:1	39	263.43	15.97	1.5

Table 7-2
FRACTAL COMPRESSION RESULTS

TEST IMAGE	INITIAL IMAGE SIZE (BYTES)	COMPRESSED FILE SIZE (BYTES)	ACTUAL COMPRESSION RATIO	COMPRESSION TIME (SEC.)	MSE	SNR (db)	SUBJECTIVE
GROUP	797,472 (624 x 426)	68,161	11.7:1	357	16.78	26.44	5
		31,907	25.0:1	340	22.84	25.10	5
		8,257	96.6:1	363	78.96	19.71	3.75
		5,345	149.2:1	600	116.45	18.03	2.5
INDIVIDUAL	797,472 (624 x 426)	79,067	10.1:1	395	38.43	21.54	5
		41,380	19.3:1	375	50.07	20.39	4.75
		7,981	99.9:1	571	234.99	13.68	2.75
		5,425	147:1	960	328.75	12.22	2
ZELDA	1,244,160 (720 x 576)	121,607	10.2:1	823	17.46	27.75	5
		62,407	19.9:1	1167	18.72	27.45	5
		11,553	107.7:1	942	51.00	23.10	4.25
		8,644	143.9:1	1500	66.50	21.94	3.75

TABLE 7-3 WAVELET COMPRESSION RESULTS							
TEST IMAGE	**INITIAL IMAGE SIZE (BYTES)**	**COMPRESSED FILE SIZE (BYTES)**	**ACTUAL COMPRESSION RATIO**	**COMPRESSION TIME (SEC.)**	**MSE**	**SNR (db)**	**SUBJECTIVE**
GROUP	797,472 (624 x 426)	77,092	10.3:1	7	152.02	16.87	5
		31,804	25.1:1	5	150.26	16.92	5
		7,823	101.9:1	4	158.97	16.67	4.25
		4,659	171.2:1	4	166.55	16.47	3.25
		1,282	622.0:1	5	377.63	12.92	1
INDIVIDUAL	797,472 (624 x 426)	74,386	10.7:1	7	194.12	14.51	5
		32,127	24.8:1	5	192.76	14.54	5
		7,804	102.2:1	4	227.50	13.82	4
		4,701	169.6:1	4	293.83	12.71	3.25
		1,269	628.4:1	5	608.22	9.55	1
ZELDA	1,244,160 (720 x 576)	121,050	10.3:1	11	30.61	25.31	5
		49,413	25.2:1	8	28.92	25.56	5
		12,340	100.8:1	7	39.18	24.24	5
		7,360	169.0:1	7	50.72	23.12	5
		2,023	615.0:1	6	121.41	19.33	2.5

The following figures contain charts comparing the results of the three compression algorithms for each test image. Figures 7.1 to 7.3 show the Signal to Noise ratio results. The 150 SNR values for JPEG and Wavelets are linearly interpolated from the 100 and 170 results so that the graphs are continuous. Figures 7.4 to 7.6 show the compression time or complexity measure. These three charts have a logarithmic Y axis (time) scale. Figures 7.7 to 7.9 show the Mean Opinion Score results. The 150 complexity and MOS values for Fractals are shown in the 170 points since interpolation could not be done for these graphs. The X axis in not linear and is used to indicate trends as compression ratio increases.

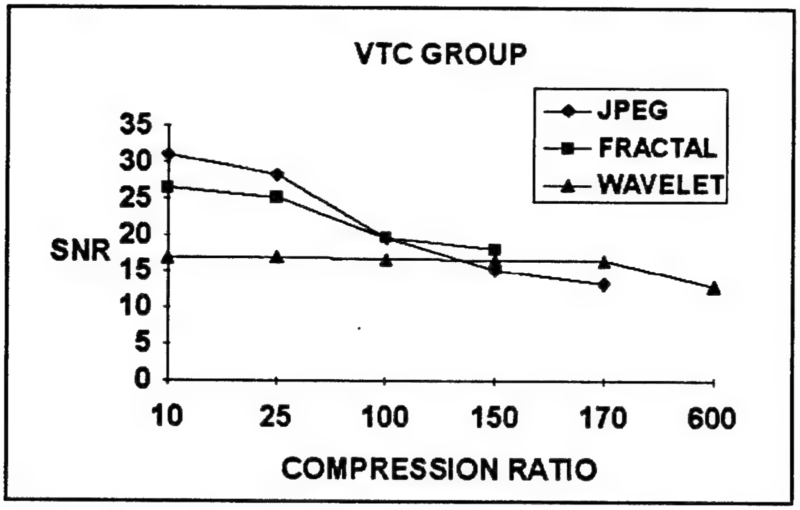


FIGURE 7.1
SNR RESULTS - VTC GROUP

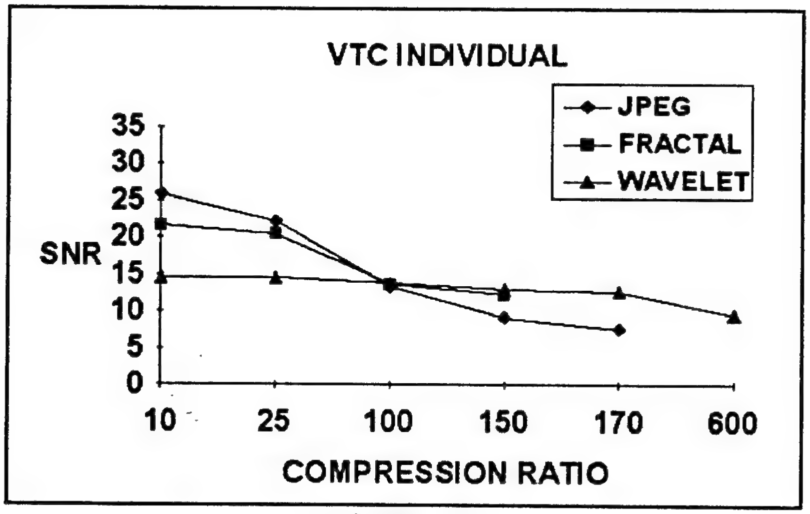


FIGURE 7.2
SNR RESULTS - VTC INDIVIDUAL

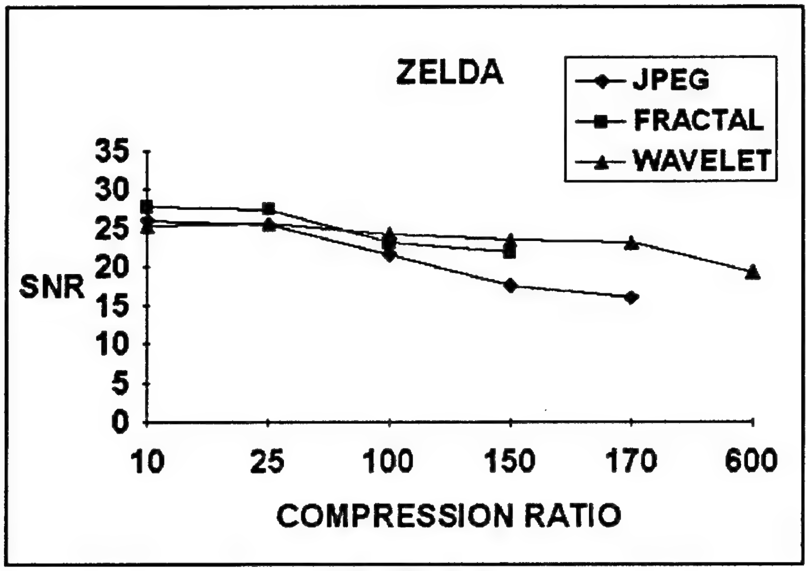


FIGURE 7.3
SNR RESULTS - ZELDA

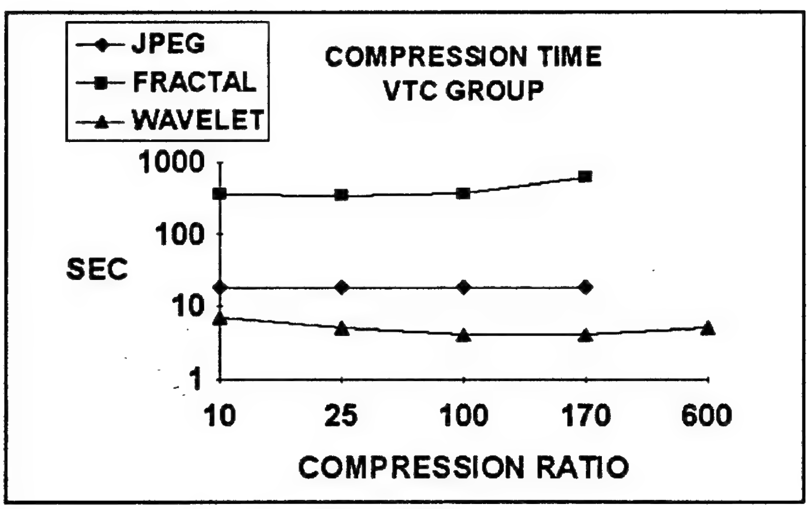


FIGURE 7.4
COMPLEXITY RESULTS - VTC GROUP

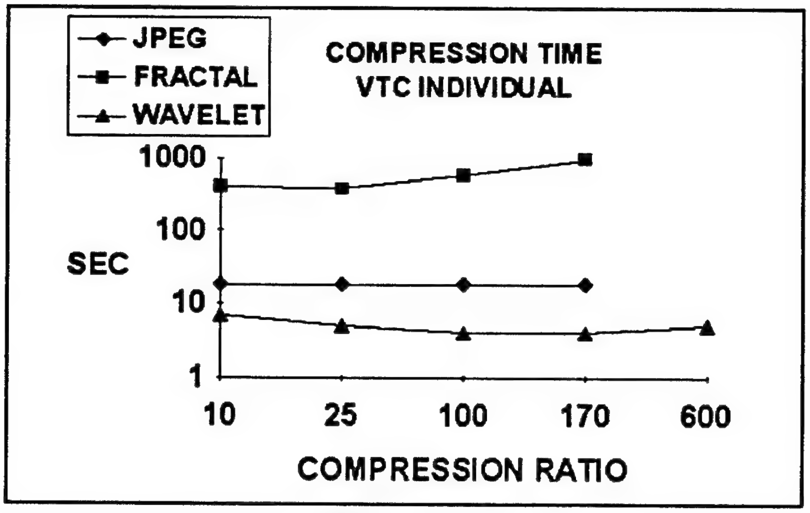


FIGURE 7.5
COMPLEXITY RESULTS - VTC INDIVIDUAL

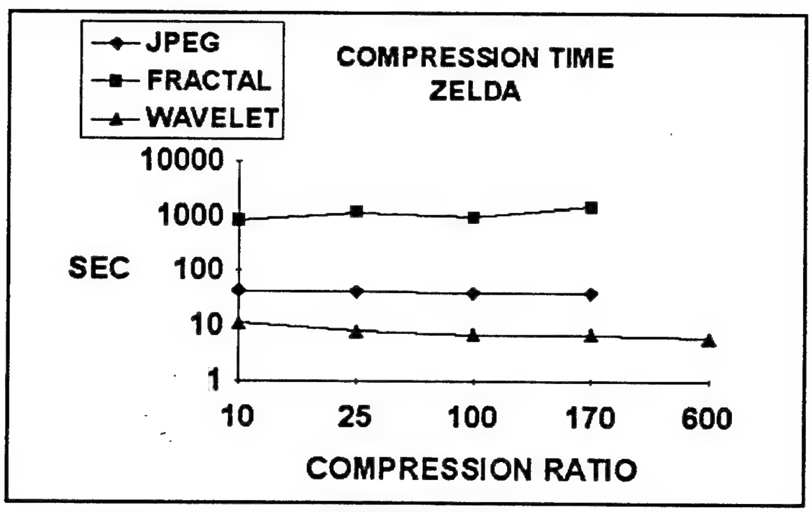


FIGURE 7.6
COMPLEXITY RESULTS - ZELDA

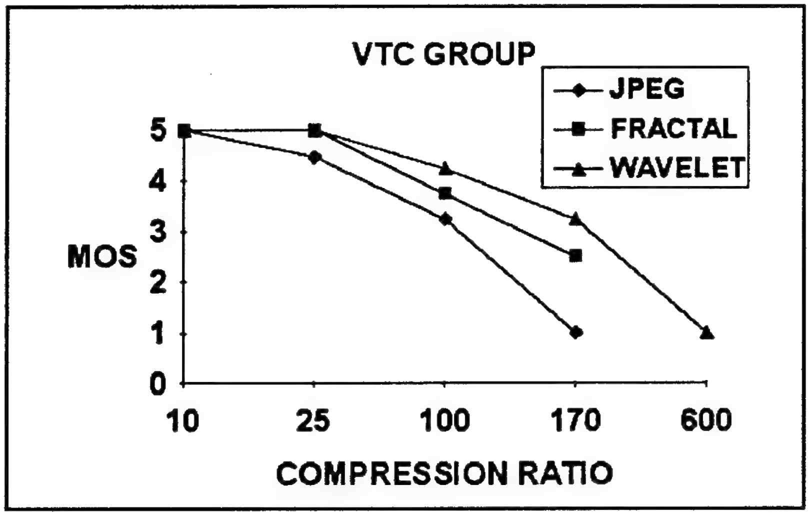


FIGURE 7.7
MOS RESULTS - VTC GROUP

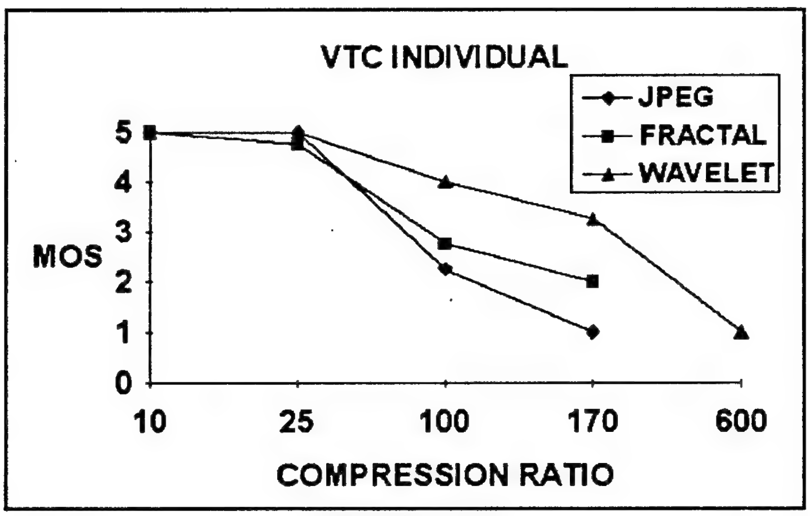


FIGURE 7.8
MOS RESULTS - VTC INDIVIDUAL

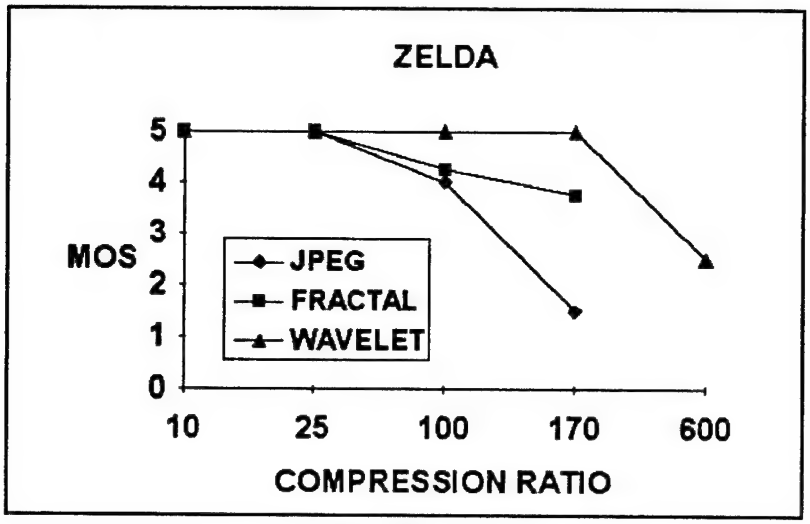


FIGURE 7.9
MOS RESULTS - ZELDA

The original test images are shown in Plates 7.1 through 7.3. There were 39 decompressed images as indicated in Tables 7-1, 7-2, and 7-3. Of these 39, 9 are shown in Plates 7.4 through 7.12. These 9 images represent the results of all three algorithms on the VTC Group test image at the compression ratios of 25:1, 100:1, and 170:1. This provides a representative sample of the performance of each algorithm at various compression levels on a typical video teleconferencing scene.

7.5 Conclusions

The three test images selected represent typical video teleconferencing scenes. The Zelda test image was to easiest to compress (i.e., better resulting quality) by all algorithms than the other images. This was probably due to the soft textures and lack of sharp edges. The VTC Individual test image was more difficult to compress for all algorithms. This was probably due to the extremely high frequency content of the image (i.e., striped shirt, books on shelf).

The Fractal algorithm could not compress the images at a ratio higher than approximately 150:1. The JPEG algorithm could not compress the images at a ratio higher than approximately 170:1. The maximum compression ratio for the Wavelet algorithm was not found, the search was stopped at approximately 600:1.

In the objective evaluation, JPEG performed the best for the two VTC test images at compression ratios less than 100:1. At ratios of 100:1 and higher Fractals performed better. For the Zelda test image, Fractals consistently performed better than JPEG at all compression ratios. The Wavelet objective evaluation was anomalous. The SNR was fairly constant for all compression ratios,

but for the two VTC test images it was significantly lower than the other two algorithms. This is surprising since the subjective quality of these images is as good or better than the corresponding images from the other algorithms. For the Zelda test image, it was also fairly constant for all compression ratios but it was also similar to the values for the other two algorithms. The cause of this anomaly has not been determined, but may be due to a shift in the decompressed image that prevents a pixel by pixel comparison.

In the complexity evaluation, the Wavelet algorithm was by far the fastest. Also, the compression time increases approximately linearly with picture size (56% increase in picture size, 57% increase in compression time). As compression ratios increase, compression times are fairly constant, however, there is a small decrease as compression ratio increases. Fractal compression was by far the slowest. It was slower than JPEG by an order of magnitude and slower than Wavelets by two orders of magnitude. Compression time increases with both increasing compression ratio and increasing picture size. As picture size increased by 56%, compression time increased by 131% to 150%. As compression ratio increased from 10:1 to 100:1 compression time increased by 1.7% to 45%. Another interesting point is that with increasing compression ratio, the compression time sometimes decreased. The JPEG algorithm has slightly slower compression times than Wavelets and they are constant for different compression ratios. As picture size increased by 56%, compression time increased by 111% to 139%.

The decompression times of the Fractal and Wavelet algorithms are very similar. Decompression time varied from about 2 to 8 seconds. Files compressed at a higher ratios decompressed faster. This is probably do to the fact that at higher compression ratios, there is less information (i.e., less coefficients) to transform back into pixels. As would be expected, Larger files decompressed in slightly longer times than smaller ones. JPEG images decompressed in 12 to 25 seconds. These results are less accurate because of the way that the programs decompressed images. For example, the Wavelet and JPEG software decompressed the images and generated output files. The Fractal software decompressed the images but did not immediately create an output file, however, it did display the decompressed image on the screen. This information is presented to give an indication of the complexity of the decoder and the symmetry of the algorithms. JPEG and Wavelets are very symmetrical in that compression and decompression complexities are equivalent. Fractals are very unsymmetrical in that the compression time is almost 100 times longer than decompression time.

In the subjective evaluation, images compressed at up to 25:1 had virtually imperceptible artifacts (i.e., visually lossless) for all algorithms. At compression ratios of 100:1, JPEG and Fractal algorithms began to exhibit artifacts but were still very usable. At 170:1 JPEG became unusable. At 150:1, Fractals had noticeable artifacts but were still usable. This is because the JPEG artifact is blocking which is very noticeable and extremely annoying, while the Fractal artifact is a loss of detail which is not constrained to blocks and is therefore less noticeable and annoying. The Wavelet quality was excellent through 170:1. At 600:1, the pictures were still usable, however, the artifacts were more noticeable. The

wavelet artifact is an oscillation around edges. In the VTC Individual image, this was more noticeable because of all of the edges.

For a video teleconferencing application, it would be reasonable to identify an MOS of 4 is the minimum acceptable for the application. That is, that distortions are slightly noticeable but not annoying. Given this criteria, a line can be drawn across each graph at MOS=4. The point at which each curve crosses the line indicates the maximum compression ratio for acceptable quality. Averaging the results over the three graphs, it can be concluded that JPEG provides acceptable quality at up to 70:1, Fractals at up to 100:1, and Wavelets at up to 150:1. The addition of interframe coding techniques could increase these results. However, it should be noted that the addition of interframe coding will probably not uniformly increase the compression of the three coding techniques.

In conclusion, the Wavelet algorithm shows tremendous promise for application to future video teleconferencing systems. Wavelets provide good quality at high compression ratios. The artifacts produced are less noticeable than the blocking artifacts produced by DCT algorithms. The algorithm is symmetrical in that the encoder and decoder are similar in complexity. The compression and decompression times are short enough that the algorithm can easily operate at real-time rates when implemented in a DSP or custom hardware.

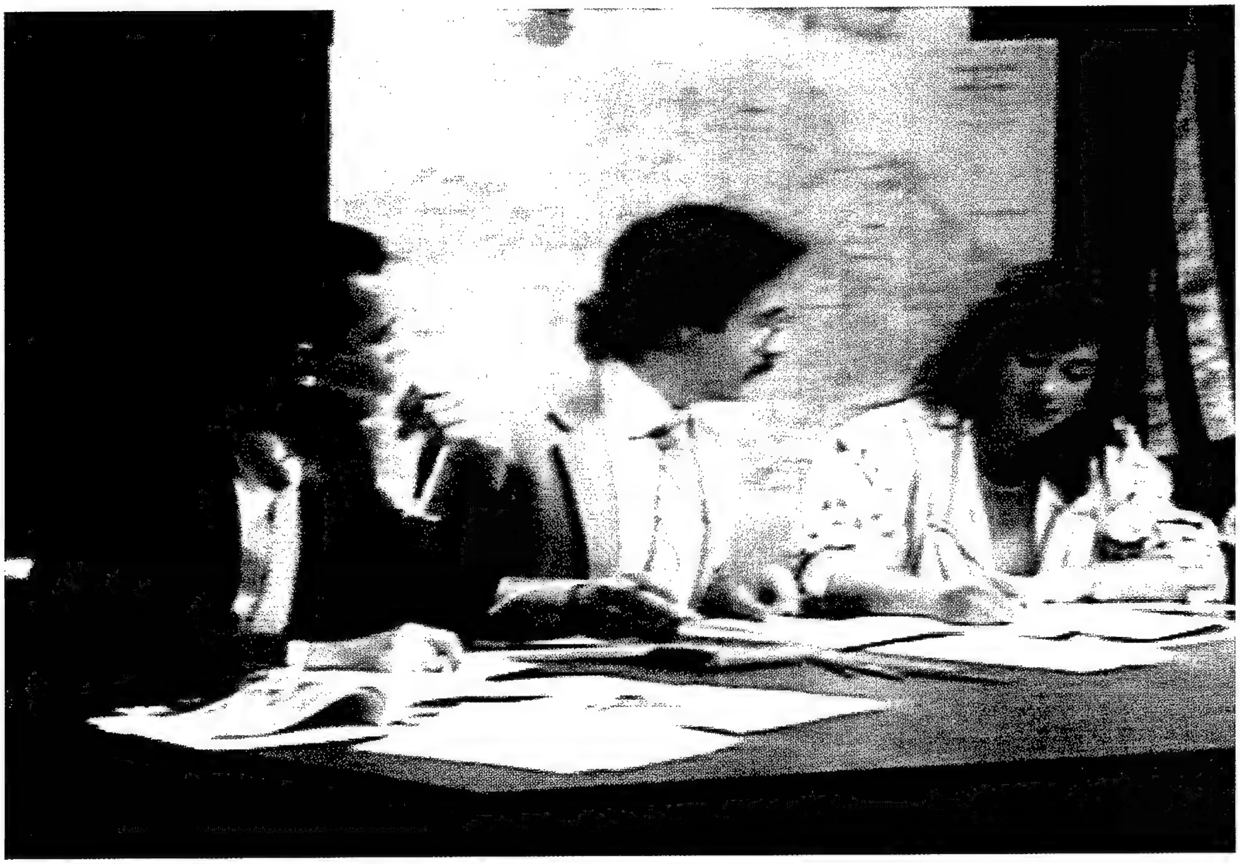


PLATE 7.1 VTC Group Test Image

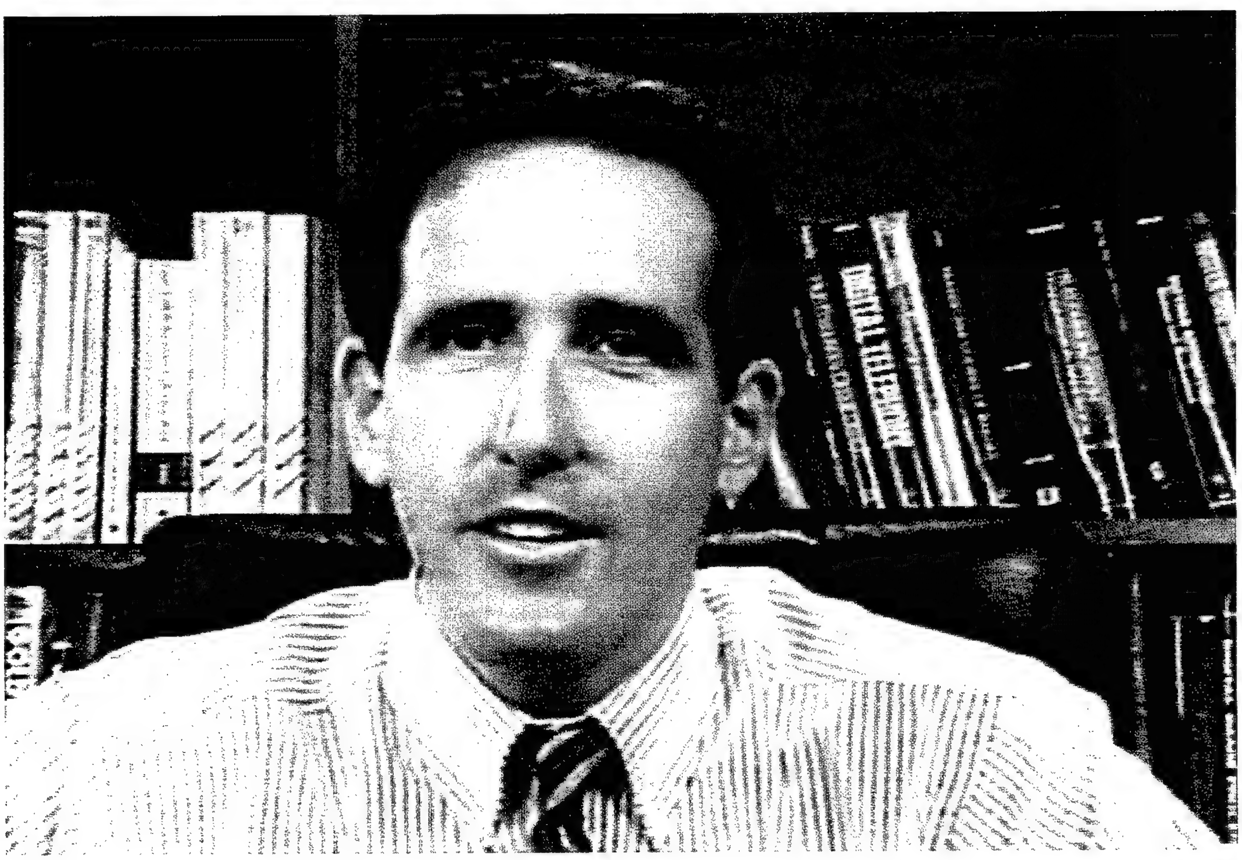


PLATE 7.2 VTC Individual Test Image

PLATE 7.3 Zelda Test Image

PLATE 7.4 VTC Group — JPEG Compressed — 25:1

PLATE 7.5 VTC Group — JPEG Compressed — 100:1

PLATE 7.6 VTC Group — JPEG Compressed — 170:1

PLATE 7.7 VTC Group — FRACTAL Compressed — 25:1

PLATE 7.8 VTC Group — FRACTAL Compressed — 100:1

PLATE 7.9 VTC Group — FRACTAL Compressed — 150:1

PLATE 7.10 VTC Group — WAVELET Compressed — 25:1

8.0 PRODUCT SURVEY

The purpose of this task is to survey commercially available video coding products which could potentially have an impact on video conferencing to enhance the existing H.261 codec standard. The six products described in Table 8-1 have been identified and are discussed below.

TABLE 8-1

COMPANY	PRODUCT NAME	CODING TECHNOLOGY	COMMENT
Iterated Systems	- POEM - POEM Videobox	Fractal	- Intraframe - Video Decode
FED-COMM International	- Matchmaker - Mapview	Fractal	Applies Iterated System Products to the Federal Government
Summus	SWIFTimage	Wavelet	Intraframe
Mac A. Cody Association	Tsuanmi Plus	Wavelet	Intraframe
INTEL	Proshare INDEO	Slant Transform	- Motion TV - Software based for PC application
Aware	AccuPress	Wavelet	Used for Multimedia and Radiology

ITERATED SYSTEMS - Iterated Systems supplies a wide range of Fractal-based products which are outlined in Appendix 8A. A brief description of six products is included in Figure 8.1. All of the products provide for the coding/decoding of still pictures with the exception of the VideoBox which decodes video. No product is available which <u>encodes</u> video in real time. The product which was used for the analysis described in Section 7.0 is Images Incorporated III.

FED-COMM INTERNATIONAL - FED-COMM International was founded to apply the Iterated System Fractal technology to the Federal Government. A package of literature describing their products is included in Appendix 8B. Verbal discussion indicates they may have a real time video encoder, but no specific literature is available.

SUMMUS - Summus supplies a Wavelet software product entitled SWIFTimage which is described briefly in Figure 8.2. It is this product which was used for the

Feature Products

Images Incorporated III: This end user product features fractal and JPEG image compression, resolution enhancement, graphics format conversion and 250 24-bit clip art images. Fractal compression is performed in the background in single or batch mode. Images Incorporated III is an OLE server and can embed fractal compression, decompression and resolution enhancement into OLE client applications. Optional FTC-III coprocessor support cuts compression time by as much as 90%. Editing functions include cut, copy, paste, rotate, vertical/horizontal flip and expand. Supported formats are FIF, FTT, TGA, BMP, GIF, PCX, RAS, TIF, and JPG.

POEM ColorBox III: This software developer's kit (SDK) includes Dynamic Link Libraries (DLLs) for compression and decompression, sample programs with source code, 20 run-time licenses and more. ColorBox III features Fractal Transform™ Template (FTT) technology and FIF Archiving to improve image quality while reducing file size by 20% or more over previous versions. ColorBox III is optimized for display-based applications like multimedia and image databases. Optional FTC-III coprocessor cuts compression time by up to 90%. ColorBox III for DOS available May 1993 (decompression only).

POEM Hi-Res 2.0: POEM Hi-Res is the flagship product of Iterated's fractal compression line. The SDK features FTT technology and FIF Archiving to improve image quality while reducing file size by 20% or more over previous versions. A range of compression options provides: total control over image quality; single and batch compression; and the option to create multiple files in one compression run. Other features include virtually unlimited input size, compression ratios exceeding 200:1, resolution independent images and near real-time, software-only decompression. The SDK includes a Windows compressor application, DLLs for integrating compression into Windows applications, and decompression DLLs and C object libraries for integrating decompression into Windows and DOS applications. This SDK is designed for extremely high resolution images and applications that require the absolute highest quality images. *Requires POEM FTC-II Compression Board.*

POEM VideoBox: This SDK contains C Object Libraries to integrate video decompression into DOS applications. Video plays at 30 frames per second (fps) in a 320 x 200 x 15-bit window (80486/33MHz) with data rates as low as 40 KB/second. Video can also play in grayscale or 8-bit mode. Sound is synchronized to individual frames with frames dropped on slower PCs to maintain synchronization.

POEM FTC-III: A compression accelerator for Images Incorporated III and ColorBox III. The latest in fractal compression hardware, FTC-III increases compression speed by a factor of three to ten times over software-only compression. Designed around the newest fractal compression chip, the FTC-III is ideal for the developers and end users who require faster compression.

POEM FTC-II: A full length 16-bit ISA standard board driven by an Intel i960 RISC processor, FTC-II has eight custom fractal compression ASICs and 1MB of on board RAM. FTC-II is Iterated's most powerful compression board and provides the highest quality compression. *Required for POEM Hi-Res.*

FIGURE 8.1 FEATURE PRODUCTS

Iterated Systems, Inc. • 5550-A Peachtree Pkwy, Suite 650, Norcross, GA 30092
(800)4FRACTL • FAX (404)840-0806 • International (404)840-0310

May 10, 1994

Summus' still image compression system, **SWIFTimage**, is intended for (lossy) compression of color or grayscale single frame raster images. **SWIFTimage** is a proprietary scheme, based on a wavelet transform combined with scalar quantization, runlength encoding, and entropy encoding of codewords.

SWIFTimage is available for Windows, OS/2, UNIX, and DOS.

SWIFTimage has the best known performance on the market today, both as far as speed of compression/decompression as well as picture quality at given compression ratio (established by independent testing, DSA 1993).

In addition to proven performance superiority, **SWIFTimage** includes a number of capabilities and features:

low memory implementation allows our algorithms to run in environments with severe restrictions on available memory (PCs, fax machines, scanners) or where memory would consume too much power (missiles, teleconferencing)

focusing/region based compression for selectively choosing areas where less compression and higher image quality is desired

magnification for quick zooming without block artifacts

enhancement for integrated image processing capabilities such as sharpening or noise removal

low contrast enhancement for quickly improving the quality of low contrast images

error correction for compensating against biterrors during transmission

fix length/variable length for allowing more error resiliency

progressive transmission for applications where it is of interest to very quickly obtain a preliminary view of the image and refine this as time progresses.

FIGURE 8.2 SUMMUS' STILL IMAGE COMPRESSION: A BRIEF DESCRIPTION

comparative analysis described in Section 7.0. Summus also supplies the source code which could be used to code motion video on an intraframe basis.

MAC A. CODY ASSOCIATES - Mac A. Cody Associates supplies a Wavelet software product entitled Tsunami Plus, and descriptive literature is included in Appendix 8C. This product is unusual because it includes the source code which could be used to code motion video on an intraframe basis.

INTEL - The DVI coding algorithm, developed originally by the Sarnoff Laboratory, was purchased by INTEL and has been expanded to form the INDEO coding product which resides in their ProShare Personal Conferencing product. Descriptive literature on this product is included in Appendix 8D. DVI, and early versions of INDEO, employed Vector Quantization coding. However, more recent INDEO products employ the Slant Transform which is simpler to implement in software than the DCT and outperforms VQ. Cost is an important consideration for INTEL since their main market thrust is for the PC platform. Since the ProShare product provides the video as a supplement to the basic exchange of computer screens, the video quality issue is probably not as critical for INTEL as it is in conventional video conferencing and videophone applications.

AWARE - Aware Technology has introduced a video coding product based on the Wavelet transform.

The results of the survey listed above tend to confirm that most of the advanced video coding work is being done in the Wavelet and Fractal areas as opposed to Vector Quantization. It also confirms some of the earlier conclusions that most of the work in Wavelets and Fractals remains focused on still pictures. No products are available which adapt these advanced techniques for interframe and motion compensation modes.

Another indication of the strength of the H.261 standard is that some codec manufacturers who previously offered proprietary algorithms as an optional alternative to the standard no longer offer these alternatives. Instead they have chosen to concentrate all their resources on the H.261 standard.

9.0 CONCLUSIONS

The ITU H.261 standard is now used in virtually all video teleconferencing systems around the world, providing a picture quality acceptable for most applications, and a high level of interoperability. In this report, advanced video coding techniques which are being investigated to possibly enhance the H.261 Recommendation have been reviewed. The four coding techniques which were examined are vector quantization, wavelets, fractals, and object-based coding. Comments on the potential for each of these techniques is provided below.

VECTOR QUANTIZATION-Vector Quantization Coding is the most mature of the four advanced coding techniques examined. Since it is a block coding technique, like H.261, it could easily use the same interframe/Motion Compensation principle as H.261. Competitive VQ systems were proposed when both the H.261 and MPEG standards were selected; DCT has always prevailed. Recent studies suggest that Wavelets and Fractals show more promise than VQ. Consequently VQ does not show promise to be the advanced techniques to improve H.261.

WAVELET CODING-Recent studies indicate that Wavelets may be the most promising advanced video coding technique. When used for intraframe coding, all indications are that it is superior to both DCT and Fractals. Unfortunately, it seems that it may be difficult to extend wavelets to interframe and motion compensation coding, which will be necessary if it is to displace H.261. At the writing of this report, there are no known systems which successfully apply wavelet coding in the temporal domain.

FRACTALS-Like wavelets, fractals show potential to outperform the DCT for intraframe coding. However, also like wavelets, it appears to be difficult to apply fractals in the interframe/motion-compensation domain. Another disadvantage for fractals is that the encoder is quite complex. Like wavelets, there are no strong fractal standard proposals, or proprietary systems in the field, which show promise of enhancing H.261 in the near term.

OBJECT-BASED CODING-Since object-based coding is fundamentally designed for interframe coding it does not suffer from the same disadvantages of fractals or wavelets. However, OBC is fundamentally very complex, and still is very much in the research stage. Researchers have reported that it will be more difficult to develop practical OBC systems than originally expected. For example, the problem of segmenting the scene into two parts--moving objects and background, has proven to be surprisingly complex.

Based upon the above viewpoints and perspectives, it would appear that the H.261 standard is not likely to be improved in the near term. It is possible that a system based on wavelets or fractals could provide a significant improvement over H.261, but this will not happen quickly. It is likely that the work on object-based coding will impact video conferencing/videophone standards in the longer term.

10.0 BIBLIOGRAPHY

10.1 General Video Coding

[1] D. Anastassiou. "Digital television," *Proceedings of the IEEE*, Vol. 82, No. 4, page 510, April 1994.

[2] R. M. Gray. *Source Coding Theory*, Kluwer Academic Publishers, Norwell, Massachusetts, 1990.

[3] A. N. Netravali and B. G. Haskell. *Digital Pictures, Representation and Compression*, Plenum Press, New York, 1988.

[4] R. Plompen. *Motion Video Coding for Visual Telephony*, PTT Research Neher Laboratories, 1989.

[5] K. R. Rao, editor. *Discrete Transforms and Their Applications*, Benchmark Papers in Electrical Engineering and Computer Science Series, Vol. 28, Van Nostrand Reinhold Company Inc., 1985.

[6] J. A. Storer, editor. *Image and Text Compression*, Kluwer Academic Publishers, Norwell, Massachusetts, 1992.

[7] J. A. Storer and M. Cohn, editors. *Proceedings, Data Compression Conference (DCC '94)*, Sponsored by IEEE Computer Society Technical Committee on Computer Communications, Snowbird, Utah, March 1994.

10.2 Motion Estimation

[8] T. S. Huang, editor. *Image Sequence Analysis*, Springer Series in Information Sciences, Vol. 5, Springer-Verlag, 1981.

[9] B. Liu and A. Zaccarin. *New fast algorithms for the estimation of block motion vectors*, IEEE Trans. Circuit and Systems, 3:2, 1993.

[10] Y. -Q. Zhang and S. Zafar, *Motion compensated wavelet transform coding for color video compression*, IEEE Trans. Circuit and Systems for video technology, 2:3, 1992.

10.3 Vector Quantization

[11] "Component vector quantization," Annex 4 of CCITT Study Group VIII, Geneva, 1-12 December 1986.

[12] H. Abut, editor. *Vector Quantization*, IEEE Acoustics, Speech, and Signal Processing Society, IEEE Press, New York, 1990.

[13] A. Gersho. "On the structure of vector quantizers," *IEEE Trans. Info. Theory* IT-28(2), pp. 157-166, March 1982.

[14] A. Gersho and B. Ramamurthi. "Image coding using vector quantization," *Proc. ICASSP*, Paris, 1982.

[15] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Norwell, Massachusetts, 1992.

[16] R. M. Gray and Y. Linde. "Vector quantizers and predictive quantizers for Gauss-Markov sources," *IEEE Trans. Commun.* COM-30(2), pp. 381-389, February 1982.

[17] J. Helden and D. E. Boekee. "Vector quantization using a generalized tree search algorithm," *Proc. 5th Symp. Information Theory in the Benelux,ri Aalten*, pp. 21-27, May 1984.

[18] H. M. Huang and J. W. Woods. "Predictive vector quantization of images," *IEEE Trans. Commun.* COM-33(11), pp. 1208-1219, November 1985.

[19] Y. Linde, A. Buzo, and R. M. Gray. "An algorithm for vector quantizer design," *IEEE Trans. Commun.* COM-28(1), pp. 84-95, January 1980.

10.4 Sub-band Coding

[20] J. W. Woods, editor. *Subband Image Coding*, Kluwer Academic Publishers, Norwell, Massachusetts, 1991.

10.5 Wavelet Coding

[21] A. N. Akansu and R. A. Haddad. *Multiresolution Signal Decomposition; Transforms, Subbands, and Wavelets*, Academic Press, Inc., 1992.

[22] C. K. Chui. *An Introduction to Wavelets*, Wavelet Analysis and Its Applications, Vol. 1, Academic Press, Inc., 1992.

[23] Y.-Q. Zhang and S. Zafar. "Motion-compensated wavelet transform coding for color video compression," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 2, pp. 285-296, Sept. 1992.

[24] M. Ohta and S. Nogaki. "Hybrid picture coding with wavelet transform and overlapped motion-compensated interframe prediction coding," *IEEE Trans. on Signal Processing*, vol. 41, pp. 3416-3424, Dec. 1993.

[25] M.L. Hilton, B.D. Jaweerth, and A. Sengupta, "Compression still and moving images with wavelets," *IEEE Trans. on Image Processing,* vol. 2, no. 3, 1994.

[26] C. K. Chui, editor. *Wavelets: A Tutorial in Theory and Applications,* Wavelet Analysis and Its Applications, Vol. 2, Academic Press, Inc., 1992.

[27] S. G. Mallat. "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. II, No. 7, July 1989.

[28] R. Neff, A. Zakhor, and M. Vetterli. *Very low bit rate video coding using matching pursuits*, SPIE Symp. on Visual Comm. and Image Proc., September 1994.

[29] R. Neff, A. Zakhor, and M. Vetterli. *Very low bit rate video coding using matching pursuits*, SPIE Symp. on Visual Comm. and Image Proc., September 1994.

10.6 Fractal Coding

[30] M. F. Barnsley. *Fractals Everywhere*. Academic Press Inc., London, 1988.

[31] J. M. Beaumont. Image data compression using fractal techniques. *British Telecom Technology Journal*, 9(4):93-109, October 1991.

[32] Y. Fisher, E. W. Jacobs, and R. D. Boss. Fractal image compression using iterated transforms. In J. A. Storer, editor, *Image and text compressiòn*, Chapter 2, pp. 35-61, Kluwer Academic Publishers, 1992.

[33] B. Hürtgen and C. Stiller. Fast hierarchical codebook search for fractal coding of still images. In *Proceedings Video Communications and PACS for Medical Applications, EUROPTO Series*, Berlin, Germany, 1993.

[34] D.M. Monro and F. Dudbridge, "Fractal approximation of image blocks," *ICASSP*, pp. 485-488, 1992.

[35] B. Hürtgen and P. Bürtgen, "Fractal approach to low rate video coding," *Proc. SPIE*, vol. 2094, pp. 120-131, 1993.

[36] B. Hürtgen and Th. Hain. On the convergence of fractal transforms. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing ICASSP'94*, 1994. In print.

[37] B. Hürtgen, F. Müller, and C. Stiller. Adaptive fractal coding of still pictures. In *Proceedings of the International Picture Coding Symposium PCS'93*, Lausanne, Switzerland, 1993.

[38] A. E. Jacquin. A novel fractal based block-coding technique for digital images. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing ICASSP'90*, Vol. 4, pp. 2225-2228, 1990.

[39] A. E. Jacquin. Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Transactions on Image Processing*, 1(1):18-30, January 1992.

[40] L. Lundheim. *Fractal signal modelling for source coding*, PhD thesis, Universitetet I Trondheim Norges Tekniske Høgskole, 1992.

10.7 Object-based Coding

[41] K. Aizawa, H. Harashima, T. Saito. "Model-based analysis-synthesis image coding (MBASIC) system for a person's face," *Signal Processing: Image Communication*, Vol. 1, No. 2, pp. 139-152, October 1989.

[42] M. Hötter and R. Thoma. "Image segmentation based on object oriented mapping parameter estimation," *Signal Processing*, vol. 15, no. 3, pp. 315-334, Oct. 1988.

[43] M. Hötter. "Optimization and efficiency of an object-oriented analysis-synthesis coder", *IEEE Trans. on Circuits. and Sys. for Video Technol.,* vol. 4, no. 2, pp. 181-194, Apr. 1994.

[44] M. Hötter. "Object-oriented analysis-synthesis coding based on moving two-dimensional objects," *Signal Processing: Image Communication*, Vol. 2, No. 4, pp. 409-428, December 1990.

[45] H. G. Musmann, M. Hötter, J. Ostermann. "Object-oriented analysis-synthesis coding of moving images," *Signal Processing: Image Communication*, Vol. 1, No. 2, pp. 117-138, October 1989.

[46] H. G. Musmann, M. Hötter, J. Ostermann. "Object-oriented analysis-synthesis coding of moving images," *Signal Processing: Image Communication*, Vol. 3, No. 2, November 1989.

[47] J. Ostermann. "Modelling of 3D moving objects for an analysis-synthesis coder," *SPIE/SPSE Symposium on Sensing and Reconstruction of 3D Objects and Scenes '90*, B. Girod, Editor, *Proc.*

SPIE 1260, pp. 240-250, Santa Clara, California, U.S.A., February 1990.

[48] H. Schiller, M. Hötter. "Investigations on colour coding in an object-oriented analysis-synthesis coder," *Signal Processing: Image Communication*, Vol. 5, No. 4, pp. 319-326, October 1993.

[49] R. Y. Tsai, T. S. Huang. *Estimating three-dimensional motion parameters of a rigid planar patch,* IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-29, No. 6, pp. 1147-1152, December 1981.

APPENDIX 2A

H.261 PICTURE STRUCTURE AND AN EXAMPLE OF DCT CODING

H.261 PICTURE STRUCTURE AND AN EXAMPLE OF DCT CODING

In the H.261 encoding process, each picture is subdivided into Groups of Blocks (GOB). As shown in Figure 2A.1, the CIF picture is divided into 12 GOB's while QCIF has only three GOB's. From the GOB level down, the structure of CIF and QCIF is identical. A header at the beginning of the GOB permits resynchronization and changing the coding accuracy.

1	2
3	4
5	6
7	8
9	10
11	12

CIF

1
3
5

QCIF

FIGURE 2A.1 ARRANGEMENT OF GOBs IN A PICTURE

Each GOB is further divided into 33 macroblocks, as shown in Figure 2A.2. The macroblock header defines the location of the macroblock within the GOB, the type of coding to be performed, possible motion vectors, and which blocks within the macroblock will actually be coded. There are two basic types of coding. In Intra coding, coding is performed without reference to previous pictures. This mode is relatively rare, but is required for forced updating, and every macroblock must occasionally be Intra coded to control the accumulation of inverse transform is match errors. The more common coding type is Inter, in which only the difference between the previous picture and the current one is coded. Of course, for picture areas without motion, the macroblock does not have to be coded at all.

1	2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30	31	32	33

FIGURE 2A.2 ARRANGEMENT OF MACROBLOCKS IN A GOB

Each macroblock is further divided into six blocks, as shown in Figure 2A.3. Four of the blocks represent the luminance, or brightness, while the other two represent the red and blue color differences. Each block is 8x8 pixels, so it can be seen that the color resolution is half of the luminance resolution in both dimensions.

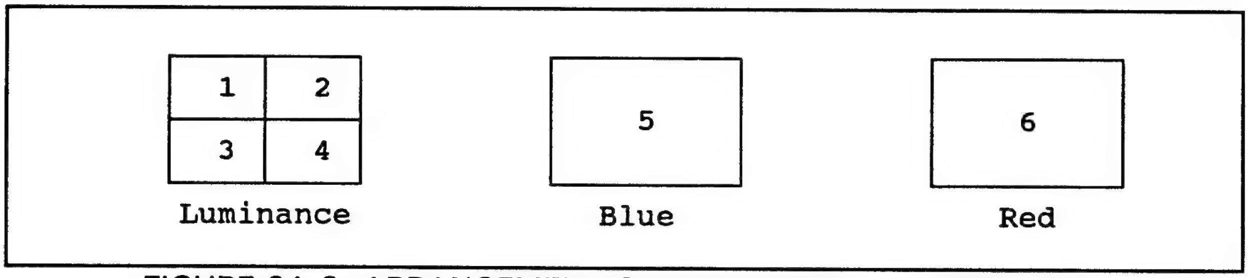


FIGURE 2A.3 ARRANGEMENT OF BLOCKS IN A MACROBLOCK

Example of DCT Coding

Figure 2A.4 shows a simple example of how each 8x8 block is coded. In this case, Intra coding is used, but the principle is the same for Inter coding. Figure 2A.4a shows the original block to be coded. Without compression, this would take 8 bits to code each of the 64 pixels, or a total of 512 bits. First, the block is transformed, using the two-dimensional Discrete Cosine Transform (DCT), giving the coefficients of Figure 2A.4b. The basis functions used for the DCT transform are illustrated in Figure 2A.5. Note that most of the energy is concentrated into the upper left-hand corner of the coefficient matrix. Next, the coefficients of Figure 2A.4b are quantized with a step size of 6. (The first term {DC} always uses a step size of 8.) This produces the values of Figure 2A.4c, which are much smaller in magnitude than the original coefficients and most of the coefficients become zero. The larger the step size, the smaller the values produced, resulting in more compression.

The coefficients are then reordered, using the Zig-Zag scanning order of Figure 2A.6. All zero coefficients are replaced with a count of the number of zero's before each non-zero coefficient (RUN). Each combination of RUN and VALUE produces a Variable Length Code (VLC) that is sent to the decoder. The last non-zero VALUE is followed by an End of Block (EOB) code. The total number of bits used to describe the block is 25, a compression of 20:1.

At the decoder (and at the coder to produce the prediction picture), the step size and VALUE's are used to reconstruct the inverse quantized coefficients, which, as shown in Figure 2A.4e are similar to, but not exactly equal to, the original coefficients. When these coefficients are inverse transformed, the result of Figure 2A.4f is obtained. Note that the differences between this block and the original block are quite small.

75	76	77	78	79	80	81	82
77	78	79	80	81	82	83	84
79	80	81	82	83	84	85	86
81	82	83	84	85	86	87	88
83	84	85	86	87	88	89	90
85	86	87	88	89	90	91	92
87	88	89	90	91	92	93	94
89	90	91	92	93	94	95	96

a) ORIGINAL BLOCK (8x8x8 = 512 BITS)

76	76	77	79	80	81	82	83
77	77	78	80	81	82	83	84
79	79	80	81	83	84	85	86
81	82	83	84	85	87	88	88
84	84	85	87	88	89	90	91
86	87	88	89	91	92	93	93
88	89	90	91	92	94	95	95
89	90	91	92	93	95	96	96

f) RECONSTITUTED BLOCK

684	-19	-1	-2	0	-1	0	-1
-37	0	-1	0	0	0	0	-1
0	0	0	0	0	0	0	0
-4	-1	-1	-1	-1	0	-1	-1
0	0	0	0	0	0	0	0
-2	0	0	-1	0	-1	0	-1
0	0	0	0	-1	-1	-1	-1
-1	-1	-1	0	-1	0	-1	0

b) TRANSFORMED BLOCK COEFFICIENTS

688	-21	0	0	0	0	0	0
-39	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

e) INVERSE QUANTIZED COEFFICIENTS

86	-3	0	0	0	0	0	0
-6	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

c) QUANTIZED COEFFICIENT LEVELS

RUN	LEVEL	CODE
0	86	01010110
0	-3	001011
0	-6	001000011
	EOB	10

TOTAL CODE LENGTH = 25

d) COEFFICIENTS IN ZIG-ZAG ORDER AND VARIABLE LENGTH CODED

FIGURE 2A.4 SAMPLE INTRA BLOCK CODING

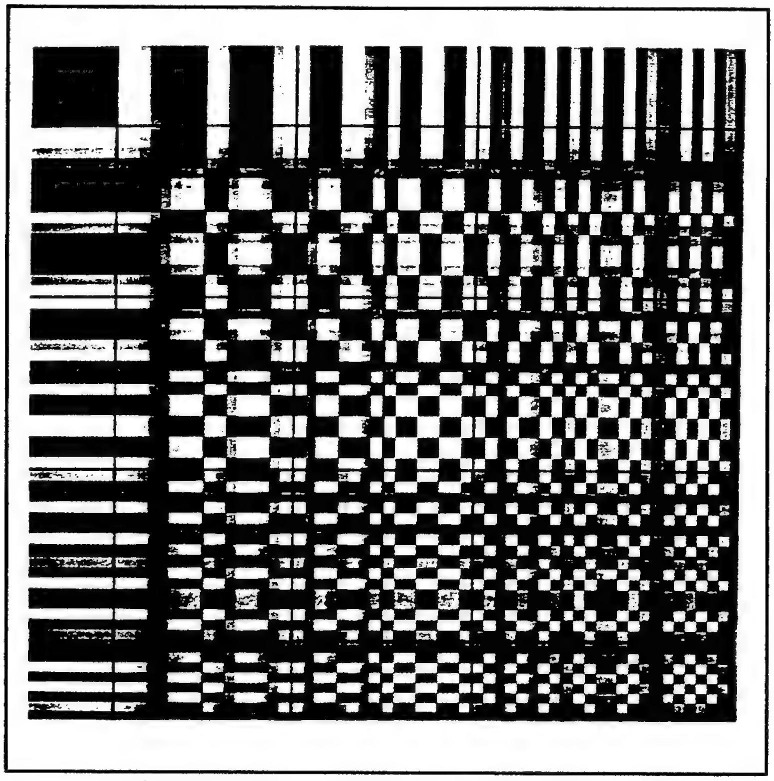


FIGURE 2A.5 BASIS FUNCTIONS OF
THE DCT AND THE WHT

1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

FIGURE 2A.6 SCANNING ORDER
IN A BLOCK

APPENDIX 8A

FRACTAL-BASED PRODUCTS
FROM
ITERATED SYSTEMS, INC.

A Picture Is Worth A Thousand Words, But Is It Worth 1 Megabyte On Your Hard Drive?

Now You Don't Have to Choose!

Digitized Images - The storage problem

Today's "image-enabled" applications have spawned a new wave of desktop imaging - from sales presentations to newsletters to databases, images are everywhere.

As much as these images enhance applications, they also quickly fill even the largest hard drive. Then you have to choose between buying a bigger drive or deleting the images you spent so much time getting on your computer in the first place. A digital "Catch-22."

Fractal Image Compression - The Solution

Fractal technology compresses images to ratios of up to 100:1, squeezing a 1 Megabyte image down to as little as 10 Kilobytes. It's long been used by multimedia and other developers to compress images for storage or transmission.

For example, Microsoft® used fractal compression to store over 10,000 images in their multimedia CD-ROM encyclopedia, Microsoft® *Encarta*. Small wonder that after reviewing **Images Incorporated III** last November, *PC Magazine* concluded "The advantages of fractal compression are unmatched."

Until recently, only developers could integrate fractal compression into applications. Now, through **Images Incorporated III**, a Windows-based product from Iterated Systems, and Object Linking and Embedding (OLE), you can save Megabytes of storage space by using fractal compression inside your current applications.

Works With Your Current Applications

How does it work? Well, you use **Images Incorporated III** to convert your image files into the compressed Fractal Image Format (FIF). Then you can delete the original image or store it off line.

Next, you embed the FIF file through commands in OLE client programs like Aldus *Persuasion*, Microsoft *Access* or Lotus *AmiPro*. That's it. The compressed FIF files are stored by the application program. When you load the document containing the FIF file, the compressed image automatically decompresses and displays just as if you had used the original bitmap.

Once the OLE link is established, operation is totally transparent, which lets you integrate compression into your applications without retraining your staff, saving Megabytes of disk space along the way.

Interested? **Images Incorporated III** costs only $295 and is available directly from Iterated Systems. Call 1-800-4FRACTL (800-437-2285) to order or for additional information.

IMAGES INCORPORATED

Iterated Systems Inc. • 5550-A Peachtree Pkwy, Ste 650 • Norcross, GA
Toll Free: (800) 4FRACTL • Fax (404) 840-

IMAGES INCORPORATED

Images Incorporated III The Essential Desktop Publishing Tool

Product Description and Specifications

Compression Options:

Fractal Transform™ Image Compression: Provides ultra-high compression, superior image quality, fast decompression and resolution independent images. Program controls include:

- *Single or batch compression*
- *Draft mode* - high quality/fast compression
- *Quality mode* - infinite quality adjustment through file size selection and 2 quality settings
- *Extended mode* - highest possible compression
- *Background compression* - Run other applications while compression is in progress.
- *FTT Technology* - improves image quality for classes of images such as fingerprints, maps, and faces.
- *FIF Archive Mode* - reduces compressed file sizes an average of 20% or more.
- *FTC-III coprocessor support* - FTC-III board reduces compression time by up to 90% compared to software-only compression. See compression table or FTC-III data sheet.
- *Decompression options* - decompress in half, full or double scale (double scale images must be 320x200 or smaller)
- *Input parameters* - 160x100 to 640x768

Image Shrink: Reduce image size in two modes:

- *Quality mode* - advanced average-down method
- *Quick mode* - subsampling

JPEG Image Compression: Fast compression and compatibility with other JPEG products. Program controls include:

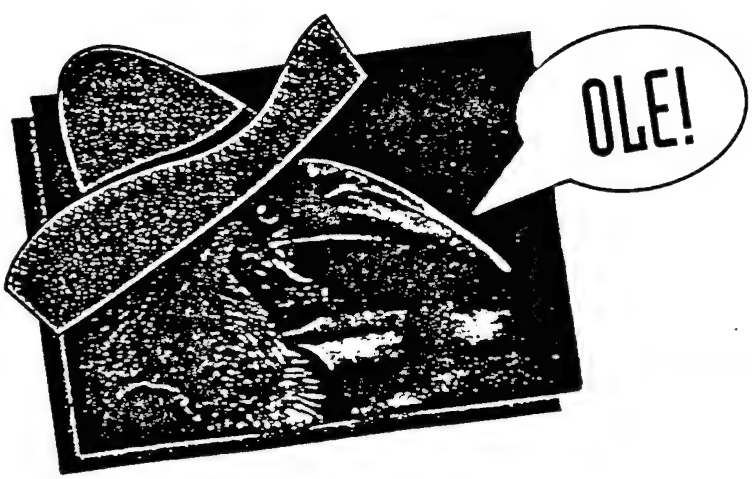
- *Quality settings from 1 - 100*
- *Store in true color or grayscale formats*
- *Compress images of unlimited size*

Object Linking and Embedding (OLE) Server: OLE Client products can access fractal compression, decompression and resolution enhancement through OLE.

Resolution Enhancement: Increase image resolution through fractal prediction. Single or batch mode operation.

- *Quick and quality modes*
- *Zoom 2, 4, 8 or 16 times original image*

Editing Functions: Cut, Paste, Rotate, Copy, Expand (pixel replication), and Vertical and Horizontal Flip.



Compression Times: Vary by processor and image size. Times for images compressed with FIF Archive Mode will be slightly higher. The following table presents a comparison of compression times for three image sizes using default compression settings over a range of compression ratios:

Image Size	CPU	Software only (min:sec)	FTC-III (min:sec)
320 x 200	80486/50	:10 - :25	:03 - :09
	80486/33	:15 - :42	:03 - :11
	80386/33	:34 - 1:36	:04 - :14
640 x 400	80486/50	1:07 - 2:04	:10 - :40
	80486/33	1:37 - 3:23	:13 - :47
	80386/33	3:48 - 7:39	:16 - 1:06
640 x 768	80486/50	2:07 - 4:10	:19 - 1:18
	80486/33	3:04 - 6:49	:25 - 1:32
	80386/33	7:50 - 15:28	:32 - 2:19

Supported Image Formats:

Format	Grayscale	Colormapped	True-Color
BMP	8-bit	256-color	24-bit
FIF	N/A	N/A	24-bit
FTT	8-bit	N/A	N/A
JPEG	8-bit	N/A	24-bit
PCX	4,8-bit	256-color	N/A
RAS	8-bit	256-color	24-bit
TGA	8-bit	256-color	24-bit
TIFF	8-bit	N/A	24-bit

Images Incorporated III automatically converts grayscale and colormapped input to 24-bit true-color before fractal compression. This enables compression of grayscale and colormapped images to 24-bit Fractal Image Format (FIF).

250 24-bit Clip Art Images: Categories include people, places, plants, animals, technology and transportation.

Minimum System Requirements:

- IBM PC or compatible 80386SX or higher
- 4 MB RAM (8 MB recommended)
- 5 MB free disk space for program and clip art library
- 5 MB permanent Windows swap file
- Windows 3.x in 80386 enhanced mode
- DOS 4.01 or higher
- Standard VGA or higher (SVGA strongly recommended)
- Mouse supported by Windows 3.x

Iterated Systems Inc.
5550-A Peachtree Pkwy.,Ste 650
Norcross, GA 30092
Toll Free: (800) 4FRACTL
Fax: (404) 840-0806
Int'l: (404) 840-0310

POEM ColorBox

Compression SDK

ColorBox III Highlights

CONTENTS:

- Compression and decompression DLLs
- A detailed manual with examples
- Example programs with source code for Microsoft C/C++ 7.0, Visual Basic and Borland C++ 3.1 for Windows
- Sample compressed and uncompressed images
- **Fractal Compression/Decompression:**
  - *Ultra-high compression* - Compress a 1.44MB 24-bit image to a 10KB FIF file. New Archive Mode further reduces file sizes an average of 20% or more
  - *Superior image quality* - FTT technology improves image quality for classes of images such as maps, faces, and fingerprints
  - *Resolution independent images* -decompress and display at arbitrary sizes up to double the original in VGA, SVGA, or true color output
  - *Fast decompression* - Decompress and display a 320 x 200 image in under 1 second on a 80486/25
  - *Background Compression* - Run other applications in foreground while compression is in progress
  - *Co-Processor Support* - Accelerate compression speed by 300% or more with optional hardware co-processor

Color Handling:

- *Automatic palette selection* - for VGA or SVGA displays; or
- *Program control of VGA or SVGA color table* - select optimal palette for display of multiple images and/or to match application interface

Simple, Comprehensive Interface:

Compression, decompression and display parameters provide program control for critical functions.

- *Compress in one of three modes:*
  - Fast mode -compress to a fixed ratio(10:1); or
  - Specified ratio - from 10:1 to 600:1 with quality controls; or
  - Specified file size - with quality controls
- *Decompress to DIB with program control over:*
  - Data source - file, buffer or array of files or buffers
  - Output to buffer or buffer array
  - Dither on or off
  - Image display resolution
  - Program controlled or optimized color table
- *Error codes documented by function*
- *Callback functions*
  - Timeslice control for background compression
  - Report progress or monitor "Cancel" control

Image Input Parameters:

- Input formats - 24-bit Windows BMP and TGA
- Panel size - compress images up to 500,000 pixels (e.g. 640 x 768 or 600 x 800) with one function call
- High resolution images - compress with multiple panels and decompress panels to shared color table with one function call for seamless display

System Requirements

Compressor:

- 80386SX or higher computer
- 4 MB RAM
- 1.44 MB floppy drive (for installation)
- Hard drive
- Windows compatible VGA, SVGA or true color video card
- Windows 3.x in enhanced mode
- 4 MB permanent swap file
- Compatible compiler such as Microsoft C/C++ 7.0 or Borland C++ 3.1

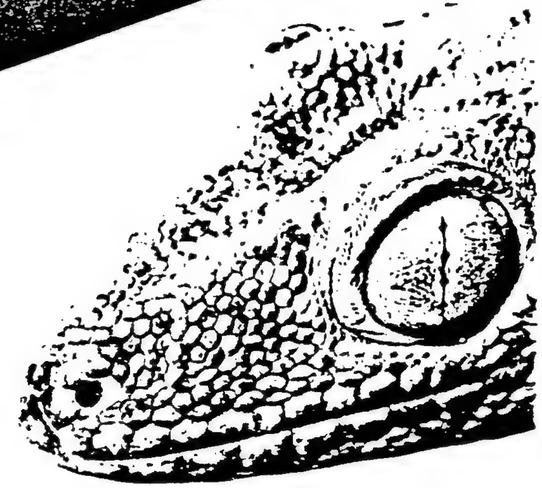
Decompressor:

- 80386SX or higher computer
- 1 MB RAM
- 1.44 MB floppy drive (for installation)
- Hard drive
- Windows compatible VGA, SVGA or true color video card
- Windows 3.x in enhanced mode
- 4 MB permanent swap file
- Compatible compiler such as Microsoft C/C++ 7.0 or Borland C++ 3.1

(System requirements for applications developed with ColorBox are identical except that a compiler is not required)

ColorBox™

A developer's kit featuring Fractal Transform™ image compression for Windows Developers



Fractal Image Compression

Advanced Technology. **POEM ColorBox III** features **Fractal Transform™** technology, discovered by Dr. Michael Barnsley. Fractal technology offers extremely high compression ratios and unmatched image quality. This means more images on your CD-ROM, diskette or hard drive, and applications that are more useful and more usable.

Fractal images are also resolution independent, and can be displayed at any resolution up to double the original size without pixelation or blockiness. This lets fractal images perform multiple functions, from thumbnail reference to full screen dis-play, and eliminates the need for multiple files.

Since fractal technology is optimized for decompression speed, there is no decompression lag. You can display a compressed 320 x 200 image in under two seconds on a 80386/33. On an 80486/33, it's under one second.

POEM ColorBox III also features sophisticated color handling to further enhance images on VGA and SVGA displays. Palettes are dynamically selected during decompression for optimum visual appeal. Or you can assign your own palette. Either way, your images look sharp, even on VGA displays. And since fractal images are inherently 24-bit, they look even better on true-color displays.

What's New? **POEM ColorBox III** debuts **Fractal Transform™** Template (FTT) technology which improves compressed image quality for classes of images with similar features or characteristics. You'll find this new technology invaluable when working with collections of images like maps, ID photographs, fingerprints, or similar categories.

Also new is FIF Archive Mode, which losslessly increases compression ratios by ten to twenty percent for all Fractal Image Format (FIF) files. Finally, **POEM ColorBox III** introduces background compression and support for the FTC-III compression co-processor which speeds compression time by at least 300%.

Up and Running in Minutes

POEM ColorBox III contains two DLLs for compression and one for decompression. It's that simple. Each has a well-documented interface that blends easily into your application. You also get sample programs with source code in Microsoft C/C++, Visual Basic, and Borland C++ 3.1 to help you get started.

We worked hard to make **POEM ColorBox III** easy because we're developers too. We know you want to shrink your images, not expand your programming task.

What the Experts Say. We think our technology is the finest in the world. But don't take our word for it. Look what others are saying.

Microsoft Multimedia Publishing Group "We selected fractal technology as the best overall mix of compression, image quality, and speed."

The Gartner Group "No other compression technology known to us approaches the unparalleled efficiency of the fractal approach."

National Institute of Standards and Technology "Unlike most compression schemes, fractal compression gives high compression with little loss of detail, yet it allows the resulting images to be decompressed to any resolution without any additional processing."

Envisioneering Times "The bottom line: ColorBox's embedded fractal technology makes color image publishing not only feasible, but cost effective."

The Bottom Line

POEM ColorBox III offers the world's best compression in an inexpensive and easy-to-use package. So call us at 1-800-4FRACTL to order or for more information, and join the growing number of developers who integrate images into their application with fractal technology.

Iterated Systems, Inc.
5550-A Peachtree Pkwy., Suite 650
Norcross, GA 30092
Tel: 800-4FRACTL Fax: (404) 840-0806
International: (404) 840-0310

POEM ColorBox III for DOS Decompression SDK

Product Description and Specifications

POEM ColorBox III for DOS allows C and C++ programmers to display highly compressed Fractal Image Format (FIF) images inside DOS applications. Designed for ease of use, POEM ColorBox III for DOS is a software developer's kit (SDK) featuring Fractal Transform™ technology.

POEM COLORBOX III FOR DOS INCLUDES:

- C object libraries for fractal image decompression
- A detailed manual
- Example programs with source code
- Sample compressed FIF files

PRODUCT HIGHLIGHTS:

- *Superior image quality* - Decompress and display images with compression ratios up to 200:1! Sophisticated color handling provides high quality images on VGA, SVGA or true color displays.

- *Resolution Independence* - Images are not restricted to original input size. Fractal Image Format (FIF) files represent images mathematically, which allows the user to display images at the optimal resolution. Decompress and display the original image at half, full, and double scale in VGA, SVGA or 24-bit output. (Double scale output is available for images with original resolution of 64,000 pixels or less. For example, a 320×200 image can be output at double scale but a 256×256 cannot.)

- *Fast decompression* - Decompression is accomplished in near real time on standard PCs without additional hardware. Decompress and display a 320 x 200 image in under 1 second on a 80486/33 in software-only.

PRODUCT FEATURES:

Decompression and display parameters provide program control over critical functions such as:

- Image Output - Images can be output to a screen, file, file array, buffer or buffer array.

- Color Handling - The POEM ColorBox III for DOS decompressor can generate an optimized color palette or can utilize a user specified color table. Display images in 8-bit colormapped or 24-bit true color.

Archive FIF file support - Decompress and display archived FIF files, which are typically 20% smaller than standard FIF files.

Fractal Transform™ Template (FTT) file support - Decompress and display files with improved quality for groups of similar images such as maps, faces, or fingerprints. FTT files use related regions within a group of images to dramatically improve quality.

Interface Compatibility - Documentation is provided for porting decompression modules from POEM ColorBox III for Windows to POEM ColorBox III for DOS.

File Compatibility - POEM ColorBox III for DOS is fully compatible with all FIF files produced by the following compressors: Images Incorporated III, Images Incorporated 2.0, POEM ColorBox III, POEM ColorBox 1.0 and POEM Color 2.1

MINIMUM SYSTEM REQUIREMENTS:

- 80386SX or higher computer
- 640 KB RAM
- Hard drive
- 1.2 MB available disk space
- 1.44 MB floppy drive for installation
- VGA or ISI supported SVGA video card
- MS-DOS 3.3 or higher
- Compatible compiler such as Microsoft C/C++ 7.0 or Borland C++ 3.1

(System requirements for applications developed with POEM ColorBox III for DOS are identical except that a compiler is not required.)

POEM ColorBox III
Compression SDK

Product Description and Specifications

POEM ColorBox III contains DLLs for integrating fractal still image compression and decompression into Windows 3.x applications. Fractal image compression is based on Fractal Transform™ technology discovered by President and Co-founder Dr. Michael Barnsley. POEM ColorBox III supplies this innovative technology to developers in an inexpensive, easy to use package.

POEM COLORBOX III INCLUDES:

- Compression: COMPENG.DLL, COMP.DLL
- Decompression: DECO.DLL
- A detailed manual with examples
- Example programs with source for Microsoft C/C++, Visual Basic and Borland C++
- Sample compressed and uncompressed images
- Twenty run-time licenses; additional licenses available

SYSTEM HIGHLIGHTS:

- **Fractal Compression/Decompression:**
  - *Ultra-high compression* - New FIF Archiving Mode reduces compressed file sizes an average of 20% or more
  - *Superior image quality* - FTT support improves image quality for groups of similar images such as maps, faces and fingerprints
  - *Resolution independent images* - Decompress and display at arbitrary size up to double the original in VGA, SVGA or 24-bit output
  - *Fast decompression* - Decompress and display a 320x200 image in under one second on a 80486/50
  - *Background compression* - Run other applications while compression is in progress

- **Color Handling:**
  - *Automatic palette selection* - for VGA or SVGA displays; **or**
  - *Program control of VGA or SVGA color table* - select optimal palette for display of multiple images and/or to match application interface

- **Simple, Comprehensive Interface:**
  Compression, decompression and display parameters provide program control of critical functions:
  - *Compress in one of three modes:*
    - Fast mode - to fixed ratio (8:1); **or**
    - To specified ratio - from 8:1 to 600:1 with quality controls; **or**
    - To specified file size - with quality controls
  - *Decompress to Device Independent Bitmap (DIB) with program control over:*
    - Data source - file, buffer or array of files or buffers
    - Output to buffer or buffer array
    - Dither on or off
    - Program controlled or optimized color table
  - *Error codes documented by function*
  - *Callback functions:*
    - Background compression
    - Monitor or cancel compression or decompression

- **Image Input Parameters:**
  - *Input formats* - 24-bit Windows BMP and TGA
  - *Panel size* - compress images up to 500,000 pixels (e.g. 640x768 or 800 x 600) with one function call
  - *High resolution images* - compress with multiple panels and decompress panels to shared color table with one function call for seamless display

SYSTEM REQUIREMENTS:

Compressor:

- 80386SX or higher computer, 4 MB RAM, hard drive, 1.44 MB floppy drive (for installation)
- Windows compatible VGA, SVGA or true color video card
- Windows 3.x in enhanced mode
- 4 MB permanent swap file
- Compatible compiler such as Microsoft C/C++, Borland C++ 3.1, or Visual Basic

Decompressor and FTT Editor:

- 80386SX or higher computer, 1 MB RAM, hard drive, 1.44 MB floppy drive (for installation)
- Windows compatible VGA, SVGA or true color video card
- Windows 3.x in standard or enhanced mode
- Compatible compiler such as Microsoft C/C++ 7.0, Borland C++ 3.1, or Visual Basic

(System requirements for ColorBox III applications are identical except the compiler is not required)

POEM VideoBox™ contains a C object library for integrating fractal video decompression into DOS applications. Fractal video is based on **Fractal Transform**™ **technology** discovered by Dr. Michael Barnsley.

POEM VideoBox™ Includes:

- A C object library for decompression
- A detailed manual with examples
- Example programs with source code for Borland C++
- Sample compressed video footage
- SVGA device drivers
- Micron BitBlaster video card

System Highlights:

Fractal Compression:

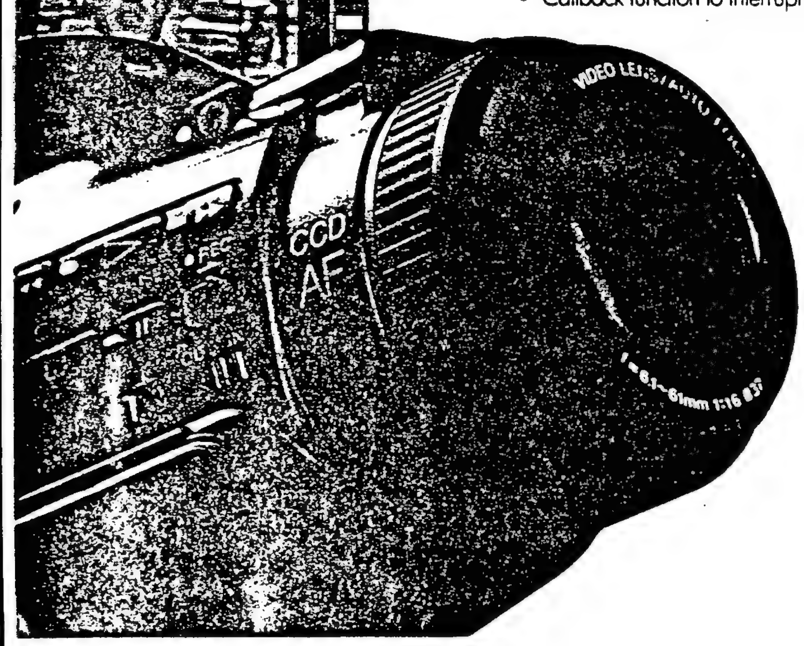
- *Professional Compression Services* - Compression performed by Iterated's skilled technicians to precise customer specifications to achieve the highest possible capture and decompression quality.
- *Ultra-high compression* - Fractal video can achieve data rates as low as 40 Kilobytes/second for high quality playback

Fractal Decompression:

- *Superior image quality* - Fractal video provides the highest subjective quality of any available software video playback system in 8 and 15-bit video mode
- *Fast decompression* - Decompress a 320x200, 15-bit video clip up to 30 frames per second (fps) on a 80486/50. Local bus accelerates performance. Slower computers will drop frames to maintain sound synchronization.

Program Controls:

- Placement of video on screen
- Playback with sound
- Grayscale, 8-bit colormapped or 15-bit high color display
- Playback speed - up to 30 fps
- Callback function to interrupt decompression



System Requirements:

Minimum:

- 80386DX/25 MHz or higher computer
- VGA or ISI supported SVGA video card
- DOS 3.3 or higher
- 640 KB RAM
- Borland Turbo C++ 3.1 compiler
- Hard drive
- 1.44 MB floppy drive (for installation)
- Sound Blaster audio card (optional)

Optimum:

- 80486/50 MHz or higher computer
- ISI supported SVGA or high color video card with 16-bit ISA bus connector
- DOS 5.0 or higher
- 640 KB RAM
- Borland Turbo C++ 3.1 compiler
- Hard drive with minimal disk fragmentation and average disk access speed of 19 milli-seconds or less
- 1.44 MB floppy drive (for installation)
- Sound Blaster audio card (optional)

Decompression Library

Description: One C object library to integrate fractal video decompression into DOS applications.

Input: Fractal Video Format (FVF) file compressed by Iterated Systems Compression Services.

Output: Full motion fractal video displaying in VGA or supported SVGA mode with optional synchronized audio. User can display video in grayscale, colormapped or high color mode.

Audio Support: Sound Blaster or Sound Blaster Pro

Decompression Speed: A 320x200 video will decompress and display on a 80486/50 up to 30 fps with audio.

(System requirements for **POEM VideoBox**™ applications are identical except the compiler is not required)

Iterated Systems

POEM VideoBox™

Seeking video technology for training, publishing or archiving and unwilling to settle for poor quality, low frame rates and shaky synchronization?

Frustrated with the choice between video co-processors that cost hundreds of dollars per station and the unacceptable quality of most software-only video solutions?

POEM VideoBox™ provides professional strength video that's ideal for virtually every video application. From full screen display to crisp, quarter screen viewing, **POEM VideoBox™** supplies the highest quality video at the industry's lowest data rates.

Introducing Fractal Video

POEM VideoBox™ is a developer's kit for integrating fractal video decompression into DOS applications. The product provides C object libraries, sample programs with source code, a detailed manual and sample videos.

POEM VideoBox™ uses patented **Fractal Transform™ technology**, discovered by Dr. Michael Barnsley, which has already spawned the finest still image compression technology available today. No less of an authority than *PC Magazine* has concluded that, "the advantages of fractal compression are unmatched."

In fact, in April 1992, the U.S. Department of Commerce and National Institute of Standards and Technology (NIST) recognized the advantages of fractal technology through a multi-million dollar research grant to Iterated Systems earmarked for video development. NIST's backing helps ensure that fractal video, already the industry's finest technology, will also become the industry standard.

"Eye-Catchingly Superior"

Fractal technology reduces video footage to "fractal" formulas to produce data rates as low as 40 KB/second. Decompression performance is the industry's best - 30 frames per second in a 320x200 window at 15-bit color depth on 80486/50 MHz computers. Frames are dynamically dropped on slower computers to maintain sound synch. **POEM VideoBox™** supports Sound Blaster, Sound Blaster Pro, and 100% compatibles.

However, the proof of the video is in the viewing, not the specs, and video quality isn't there the numbers are meaningless. Quality is wher **POEM VideoBox™** really excels.

Bill Caffery, Vice President and industry analyst at Gartner Group Inc Stamford CT, tracks compression technology for his clients, which inc some of the world's largest corporations. Frequently quoted in indust magazines like *PC Week* and *InfoWorld*, Bill is intimately familiar wi software video market. Here's what he has to say about **POEM VideoE**

"Seeing is believing. Clearly, in resolution, bit depth, frame rate and sound-synch precision, **POEM VideoBox™** outperforms Video for Wir and Quicktime. But even that edge pales once you experience **the a** free video quality of fractal technology. It's absolutely eye-catchingly superior."

POEM VideoBox™ wins the "triple crown" of software-only video - **lo** bandwidth; best performance; and highest quality. That's **why we c** professional strength video.

All This and Accessible Too!

After shipping developer's systems for 5 years, we've learned that su technology isn't enough - it has to be accessible. The **POEM VideoB** application interface was designed for ease of integration and **is ext** well documented. We also include demonstration programs (with sc code) to make sure you're up and running in minutes, not hours. **Lik** Iterated's developer's products, **POEM VideoBox™** is supported by c dedicated team of technical support representatives.

The Bottom Line

Video can be an invaluable presentation or communications tool, bu when it looks as good as your application. If professional strength v what you're looking for, **POEM VideoBox™** is the answer.

Call 1-800-4FRACTL for more information.

POEM VideoBox Decompression SDK

Product Description and Specifications

POEM VideoBox contains a C object library for integrating fractal video decompression into DOS applications. Fractal video is based on Fractal Transform™ technology discovered by Dr. Michael Barnsley.

POEM VIDEOBOX INCLUDES:

- A C object library for decompression
- A detailed manual with examples
- Example programs with source code for Borland C++
- Sample compressed video footage
- SVGA device drivers
- Micron BitBlaster Video Card

SYSTEM HIGHLIGHTS:

Fractal Compression:

- *Professional Compression Services* - Compression performed by Iterated's skilled technicians to precise customer specifications to achieve the highest possible capture and compression quality.
- *Ultra-high compression* - Fractal video can achieve data rates as low as 40 Kilobytes/second for high quality playback.

Fractal Decompression:

- *Superior image quality* - Fractal video provides the highest subjective quality of any available software video playback system in 8 and 15-bit video mode.
- *Fast decompression* - Decompress a 320 x 200 up to 30 frames per second on a 80486/50. Local bus accelerates performance. Slower computers will drop frames to maintain sound synchronization.

PROGRAM CONTROLS:

- Placement of video on screen
- Playback with sound
- Grayscale, 8-bit colormapped or 15-bit high color display
- Playback speed - up to 30 fps
- Callback function to interrupt decompression

SYSTEM REQUIREMENTS:

Minimum:

- 80386DX/33 MHz or higher computer
- VGA or ISI supported SVGA video card
- DOS 3.3 or higher
- 640 KB RAM
- Borland C++ 3.1 compiler
- Hard drive
- 1.44 MB floppy drive (for installation)
- Soundblaster audio card (optional)

Optimum:

- 80486/50 MHz or higher computer
- ISI supported SVGA or 15-bit color video card with 16-bit ISA bus connector
- DOS 5.0 or higher
- 640 KB RAM
- Borland C++ 3.1 compiler
- Hard drive with minimal disk fragmentation and average disk access speed of 15 milliseconds or less
- 1.44 MB floppy drive (for installation)
- Soundblaster audio card (optional)

(System requirements for POEM VideoBox applications are identical except the compiler is not required)

Decompression Library

Description: One C object library to integrate fractal video decompression into DOS applications. Library is compatible with Borland C++ 3.1 compiler.

Input: Fractal Video Format (FVF) file compressed by Iterated Systems Compression Services.

Output: Full motion fractal video displaying in VGA or supported SVGA mode with optional synchronized audio. User can display video in grayscale, colormapped or high color mode.

Audio Support: SoundBlaster or SoundBlaster Pro

Decompression Speed: A 320 x 200 video will decompress and display on a 80486/50 up to 30 frames per second (fps) with audio.

Video Modes: The following video modes are supported for direct screen output:

Video Mode	Description
VGA 13H	320 x 200, 256 colors
SVGA 100H (VESA standard)	640 x 400, 256 colors
SVGA 101H (VESA standard)	640 x 480, 256 colors
SVGA 103H (VESA standard)	800 x 600, 256 colors

Super VGA Drivers: The Super VGA drivers listed below are provided with VideoBox. This list may change without notice; please contact your sales representative for details.

Video Card	ISI Driver Name
ATI VGAWONDER (ATI18800 chipset)	ISIATI88.SYS
Diamond SpeedStar 24X	ISI24X.SYS
Micron BitBlaster, Boca Super Accelerator VGA (Cirrus Logic GD5426 chipset)	ISICIRUS.SYS
Paradise PVGA1A, PVGA1B (WD90C00) chipset	ISIPRDS.SYS
Trident 8900 VGA chipset	ISITRI89.SYS
Tseng Labs ET4000 chipset	ISITSNG4.SYS
VESA (generic)	ISIVESA.SYS
Video Seven VRAM, VRAM II, and VGA1024i cards	ISIVID7.SYS

The following table shows which SVGA (VESA) graphics modes are supported on various SVGA cards:

Video Card	SVGA (VESA) Modes
ATI VGAWONDER	100h, 101h, 103h
Diamond Speed Star 24X	100h, 101h, 103h, 10dh, 110h, 113h
Micron BitBlaster, Boca Super Acccelerator VGA (Cirrus GD5426 chipset)	100h, 101h, 103h, 10dh, 110h, 113h
Paradise PVGA1A, PVGA1B(WD90C00 chipset)	100h, 101h, 103h
Trident 8900 chipset	100h, 101h, 103h
Tseng ET4000 chipset	100h,101h, 103h, 10dh, 110h, 113h
VESA (generic)	100h, 101h, 103h, 10dh, 110h, 113h
Video Seven VGA1024i	100h, 101h
Video Seven VRAM II	100h, 101h, 103h
Video Seven VRAM VGA	100h, 101h, 103h

NOTE: Not all supported SVGA cards will support all the listed modes. Modes 101h, 202h, and 103h require 512K of display memory. Mode 105h requires 1M of display memory.

Supported Video Modes: The following list gives properties of the video modes supported for output by VideoBox.

Video Mode*	Bits/ Pixel	Screen Resolution	# of Colors	Graphics Cards
13h	8	320x200	256	VGA, SVGA
100h	8	640x400	256	SVGA
101h	8	640x480	256	SVGA
103h	8	800x600	256	SVGA
10dh	8	320x200	32K	SVGA
110h	8	640x480	32K	SVGA
113h	8	800x600	32K	SVGA

***NOTE:** SVGA modes 100h through 113h are defined by the Video Electronics Standards Association (VESA).

APPENDIX 8B

PRODUCT LITERATURE
FROM
FED-COMM INTERNATIONAL, INC.

FED-COMM
International

FED-COMM INTERNATIONAL, INC.
TECHNOLOGY OVERVIEW

FED-COMM International, Inc. is a small business, woman-owned company located in Arlington, VA providing the most advanced fractal image compression products enabling the successful implementation of image processing in a resolution independent manner at ultra-high compression ratios. FED-COMM's fractal compression products are based on the patented Fractal Transform Encoding technology.

Fractal compression is a revolutionary technology that incorporates several unique benefits and significant attributes:

FRACTAL TRANSFORMATION ENCODING: The encoding process converts a pixel image to a seamless mathematical algorithm which can be decompressed to resolution and color depths both higher and lower than the original image. This "resolution independence" allows for compressed images to be "future proof" since fractal images can be displayed on increasingly higher resolutions as display technologies evolve. FED-COMM's high speed fractal compression board (the RTC -100) which incorporates a frame grabber, fractal encoders and the CAIS (Content Addressable Image Search) system allows for the only real-time video processing over a single telephone line available today.

COMPRESSION AND ZOOM: Both still and video images (30 frames/second) can be compressed with complete retention of intelligence value at ratios far greater than obtainable with JPEG/MPEG. Following decompression, an image or part thereof can be expanded 2-8 to 1 with retention of intelligence value. Anti-aliasing occurs without the use of sophisticated techniques since the image is re-rendered larger rather than undergoing pixel expansion. Compressed color video (with sound) can be decoded and played back at CD-ROM speed with software alone.

Image source can be still photographs, negatives, video, air recon or surveillance film, or output from CCD, IR or SAR. The high compression ratio permits greater efficiency in transmission, temporary storage, archiving or further processing. Use of this software does not preclude use of other analytical or enhancement techniques.

SUBJECT/OBJECT IDENTIFICATION: The CAIS system is an automated image identification system which utilizes the Fractal Encoding data to analyze large images (e.g. air recon film or crowded surveillance stills or video) or sets of images (e.g. mug shots or watch lists) for the presence of a corresponding key. The key can be a face, a vehicle or other objects or parts thereof. Multiple keys (e.g. face frontal and profile views, characteristics components of an object) can be used at the same time. The CAIS system is sensitive to geometry, color, radiometry and texture. The CAIS system integrated with our RTC-100 fractal compression board will perform real time video compression ratios ranging from 150-250 to 1 with playback at rates of 30 frames per sec.

FED-COMM INTERNATIONAL invites you to further examine our complete line of revolutionary imaging products.

2000 North 14th Street
Suite 310
Arlington, VA 22201

Tel: 703/524-0063
Fax. 703/524-6483

MAPVIEW MAKER PLUS

The MAPVIEW MAKER PLUS is a complete production system which compresses digital map files at ultra-high compression ratios for viewing with MAPVIEW EXPLORER software.

MAPVIEW MAKER PLUS includes a full-size AT-compatible board that utilizes the powerful Fractal Transform process to compress digital maps at ultra-high compression ratios.

The board's modular design includes eight custom-designed Fractal Transform ASIC chips, a fast Intel i960 processor, and a high-speed static RAM. The board also features 1 MB of memory.

When compressed by the board, digital map files become Fractal Image Format (FIF) files. These FIF files are displayed by MAPVIEW EXPLORER software. MAPVIEW EXPLORER enables the user to pan maps and zoom into specific areas of maps to see greater detail. (A copy of MAPVIEW EXPLORER is included with MAPVIEW MAKER PLUS.)

The board requires a digitizing scanner or frame grabber for input and supports 24-bit, true-color Targa input files. The input file size is limited only by your PC's memory. The MAPVIEW MAKER PLUS includes a utility which splits input files into manageable 640 x 400 bits/pixel panels to facilitate compression. The MAPVIEW EXPLORER realigns these panels for display.

The board features user-selectable compression ratios and compression times. Compression time varies depending upon the size of the map being compressed, the compression ratio desired, and the compression speed selected. Fast compression producing a smaller file size is well suited for editing and browsing through large map databases Longer compression time and higher compressed file sizes provide image quality at near perfect original rendition.

The MAPVIEW MAKER PLUS system reduces disk storage space and facilitates the sharing of maps across networks or communications lines.

Hardware Specifications for Board

Full-size AT compatible board
25 MHz Intel 80960 RISC CPU
High-speed static RAM
Eight Fractal Transform ASICs
LED Diagnostics

Operating Requirements for Board

80386 or 80486 with full-size AT slot
VGA graphics adapter with monitor
3.5 A DC (max) power available
0 to 70 C operating temperature
Hard drive (minimum 40 MB recommended)
4MB extended RAM
DOS 3.0 or higher

Operating Requirements for MAPVIEW EXPLORER

80386 or 80486
4MB of extended RAM
VGA or SVGA card
a Microsoft-compatible mouse
DOS 3.0 or higher

Product specifications subject to change without notification.

MATCHMAKER FOR WINDOWS

DESCRIPTION

MatchMaker is a unique product based on the patented Fractal Transform technology. MatchMaker compares a given image (the "image key") with a collection of images or image regions (the "templates") and provides its own answers to such questions as "Which template image is most like the image key?" or "Which images are most likely to contain the image key and where?" MatchMaker is looking for imagery of similar information content, not merely for exact duplicates of the key image. MatchMaker encodes the key and templates imagery using Fractal Transform technology, analyzes resulting fractal codes and produces a list of numerical scores which are termed "match scores" or "relevance rankings". A successful MatchMaker run results in higher scores being associated with those template images which human observers would rank as most similar.

This Windows product operates in a software-only mode or can be accelerated with the RTC-100 board produced by FED-COMM International. A 480 x 480 image can be processed in as little as 7 seconds with this application. Results are provided in files and on screen. Screen display show both a numerical measure of relative similarities and regions of relative similarity in imagery.

Sample Applications

Pre-Screening-Intelligence

An image analyst needs to classify aircraft in 1000 images but only has time to look at imagery representing 10. Pre-screening is used to select 1% of the imagery most likely to contain aircraft.

Identification-Law Enforcement

Create a digital "watch list" of known criminals and store on a portable computer for site investigators. Using an artist's sketch from on site investigators at the scene of a crime, have MatchMaker select those watch list photos most likely to match the artist's sketch.

Verification-Arms Control

Examine high resolution satellite photos for detection of non-compliance with international treaties.

Security-Personnel

Digitize real time video of personnel/fingerprints and establish/verify the identity with on-line file photos/fingerprints.

Classification

Classify complex image types such as ships, vehicles, geological images, military equipment, ect.

Requirements

Operating System:	Microsoft Windows 3.1 and DOS 5.0 or higher
Computer:	Intel 80386 or 80486 or compatible
RAM:	8 Megabytes minimum, 16 Megabytes recommended
Speed:	33 MHz minimum, higher recommended
Graphics:	SVGA with Windows 3.1 drivers
Floppy:	3.5"
Hard Drive:	20 Megabytes free space when running Windows 3.1, minimum
Input:	8 bit or 24 bit digital imagery in TGA or BMP format
Accelerator:	FED-COMM International's RTC-100, optional

Performance

The following performance measures were obtained from a Toshiba T6400C with 12 Megabytes of RAM, approximately 100 Megabytes of free hard disk space, 50 MHz clock doubled Intel 80486 central processing unit and an active matrix display. The T6400C was equipped with a RTC-100 accelerator board.

MatchMaker produces a "match score" for each given image key and image template. Two performance measures are provided. One measure of performance is Total Time. MatchMaker offers the display of an internal clock to measure execution time to the nearest second. Total Time includes overhead for setup as well as time spent in matching. A second performance measure is Matches/Second, computed by dividing the Total Time by the number of templates. The performance measures are dependent on the size of the image key, the site of the template images and the number of template images. The match times also vary with the image content. The times below are for a relatively busy, technical intelligence type image of an aircraft decommissioning facility of resolution 484 x 480 x 8 bits/pixel.

Image Key	Template Size	# Templates	Total Time (seconds)	Matches/Second
16 x 16	16 x 16	900	51	17.6
16 x 16	32 x 32	225	13	17.3
16 x 16	64 x 64	49	7	7.0
32 x 32	16 x 16	900	52	17.3
32 x 32	32 x 32	225	17	13.2
32 x 32	64 x 64	49	12	4.1
64 x 64	16 x 16	900	72	5.2
64 x 64	32 x 32	225	47	4.8
64 x 64	64 x 64	49	32	1.5

Restrictions: The image dimensions must be at least 16 pixels on a side. If M is the Maximum of the width of the key image and twice the template image, and if m is the sum of the height of the key image and twice the height of the template image, then m * M must be less than 32K. MatchMaker contains a feature which automatically tiles large images into smaller images so that the 32K restriction can be satisfied. The system has been tested with 2000 template images and template images as large as 1K x 1K. However, the normal DOS and Windows; restrictions on size and numbers of directories and files apply.

APPENDIX 8C

PRODUCT LITERATURE
FROM
MAC A. CODY ASSOCIATES

Introduction

This is the User's Manual for Tsunami Plus. Tsunami Plus is an ANSI/ISO C source code library of wavelet transform routines and other support functions. The library's name, **Tsunami Plus**, was chosen because *tsunami* means, in Japanese, a tidal wave usually caused by an earthquake. The wavelet transform surged onto the scene in the last eight years and has generated a great deal of interest in the mathematics, scientific, and engineering communities. The *plus* indicates that the library is more than just a collection of transform routines. Support routines are provided to allow the programmer to quickly develop DOS applications with VESA VBE graphics support (Note: VESA and VBE are trademarks of the Video Electronics Standards Association). Not only can wavelet transforms be executed; their input parameters can be modified and displayed with a menuing system and the data can be displayed in various formats.

The intent of this library is to make wavelet technology accessible to the software development community. There is now an abundance of papers and books explaining the theory and providing the proofs to support wavelets. Up until now, there has been a decided lack of materials and tools to bridge the gap from those theoretical concepts to the practical aspects of implementation and use of this new technology.

Tsunami Plus provides a "toolbox" approach to the use of wavelets in your applications. The library is design so that only those components actually required by your applications need to be compiled and linked to them. Excess "baggage" is minimized.

The sections of this manual are designed to work together in making the Tsunami Plus library easy to use in your applications. The remainder of this section provides information on product support (in case you require assistance) and installation procedures (so you can get started). The following section, **A Practical Tutorial on Wavelets**, is a forum for introductory information on wavelets. The aim of this section is to provide a basic understanding of wavelets and wavelet transforms while avoiding much of the mathematics used to derive them. The third section, **Using Tsunami Plus**, provides information on compiling and linking the source code into your applications, porting the library to other compilers/platforms, and the demonstration programs provided with the library. The source code for the demo programs provide useful insight into applying the various routines in the library. Information on each of the routines in the library is provided in section four, **Library Reference**. Section five is the **Appendices**, which contains various useful materials such as the License Agreement and Limited Warranty.

Finally, the artwork on the cover is a reproduction of the classic Japanese print *Behind the Waves off Kanagawa* (also known as "The Great Wave") by Katsushika Hokusai (b. 1760 - d. 1849). It is part of his collection *Thirty-six Views of Mt. Fuji* (In this scene, Mt. Fuji is dwarfed in the lower center). This image was selected because of the obvious connection to tsunamis (a visual pun, if you will) and the fact that it is a beautiful print! The Tsunami Plus logo contains the kanji characters (reading from left to right) *tsu* (port) and *nami* (wave). Tidal waves usually have their greatest impact when they come to into port.

Product Support

Mac A. Cody Associates provides several means for answering your questions about Tsunami Plus. Comments and suggestions for additions and improvements to Tsunami Plus are also welcome. To contact us via mail, please send your correspondence to:

Mac A. Cody Associates
1901 N. Waterview
Richardson, TX 75080

Please allow time for a response to your correspondence, especially if you are overseas. For quicker response, support can be obtained via electronic mail. If you subscribe to America Online®, you can reach us by addressing your email to **Mcody**. If you can access the Internet, we can be reached by addressing your email to **mcody@aol.com**.

Whether you use standard mail or email, please explain your question(s) in detail. Too much detail is preferred over not providing enough. If it helps, include a brief code listing or illustration to support your question(s). Also, please provide the version and serial numbers on your Tsunami Plus distribution diskette, the compiler brand and version number, and the computer type and operating system version. We will try to be prompt in answering all questions.

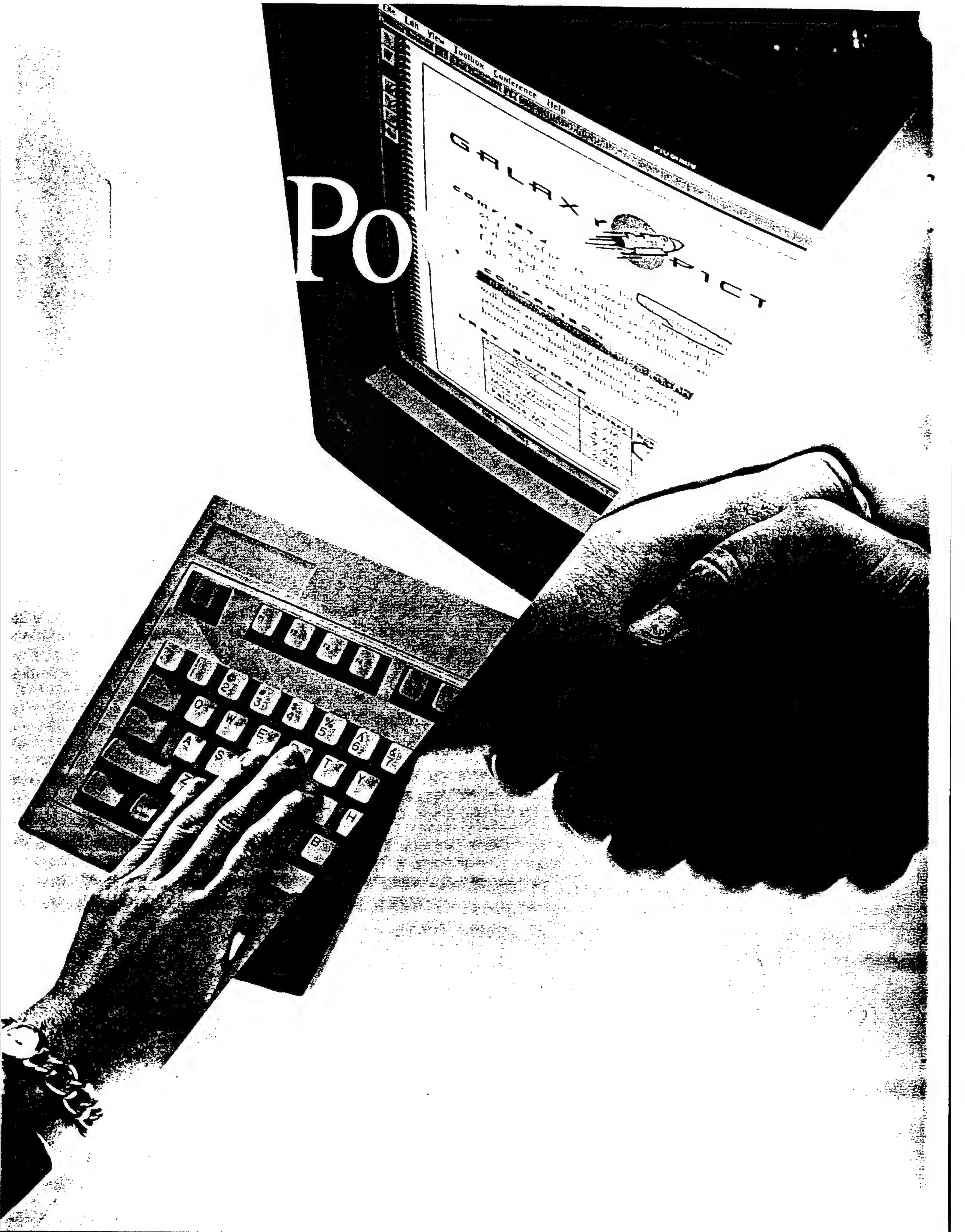
Lastly, we can be reached by telephone at (214) 437-5761. When you call, pleas be prepared to provide the serial number on your Tsunami Plus distribution diskette, the compiler brand and version number, and the computer type and operating system version. In the future, a bulletin board system is planned to provide further support. An announcement will be made when this service becomes available.

APPENDIX 8D

PRODUCT LITERATURE
FROM
INTEL PROSHARE

Intel ProShare™ Personal Conferencing

PC Power.
One-on-One
Communications.


Po
File Edit View Toolbox Conference Help

Intel ProShare™ Personal Conferencing

Brrrmnng! Your PC rings. It's Fred. He looks concerned. Doesn't understand part of your presentation. You take a look together, have a quick discussion, and Fred signs off, happy as a clam.

The boss calls. She wants to go over your budget NOW. You pull it up on your shared on-screen notebook, connect across the network, and both mark changes and highlight key points, simultaneously. When you're done, she adds it into her budget and moves it up the line.

Visions of office life in the future? No, they're the kinds of things you can do today using Intel's ProShare™ Personal Conferencing products.

Meet one-on-one even when you're far apart. Share ideas freely. Make decisions faster. Build consensus among dispersed teams more effectively. And improve communications.

It's all possible with a simple enhancement to your personal computer. Your PC already contains most of the information you discuss with others. It can edit and calculate as well as communicate. It's familiar, easy to use, and just about everybody has one. Add personal conferencing to the mix and you have an incredibly powerful communications tool right on your desk.

Intel's family of personal conferencing products extends the power of your PC to help you communicate better, and with more clarity. Start with the ProShare product you need now, add new capabilities later. All ProShare products work together. They also work with the rest of your PC: hardware, software applications and communications standards. So you don't have to learn anything new; just build on what you already know.

Doing business today is all about communicating more effectively, building consensus quickly, making decisions faster, and getting there before your competition. With ProShare Personal Conferencing products on your PC, you'll be working one-on-one. Anytime, anywhere.

One-on-One Communications

Intel ProShare

Intel ProShare™ Personal Conferencing so

Share ideas and information right here, ri

With Intel's ProShare Personal Conferencing software, you can connect your PC with a colleague's and share any Windows-based documents simultaneously. It doesn't matter if you're across the building or across the country. All you need is a modem or LAN link, and you'll be communicating more clearly and effectively than you ever thought possible!*

You're here. They're there. Sharing the same screen.

Communicating via ProShare software is unbelievably easy. On the phone you talk. On the PC you work in an on-screen notebook with tools just like those on your desk, so everything's completely intuitive. When you want to share a document with someone else in another location, first take a "snapshot" of it, marking the portion you want to discuss or importing the entire document as a print capture. Making the connection to your colleague's PC is as easy as clicking on his or her name in the call list. Once connected over a modem or LAN, the document appears on both your screens—instantaneously!

Now grab one of your electronic tools—choose from pens, highlighters and pointers—and mark up the document as you discuss it. Circle an item you want your colleague to notice. Highlight a key item. Point to a graphic element you'd like to discuss. Type in new wording to see what works best. You can even type private notes to yourself in a separate part of the notebook only you can see.

If your application supports OLE (Object Linking and Embedding), either of you can click on the object in the notebook, launch the source application, and instantly update the snapshot. When you're done, save your session for review later. What used to take hours now takes minutes!

Easy to learn, easy to use.

ProShare Personal Conferencing software has a friendly, familiar Windows-based interface that lets you quickly point and click your way around your shared on-screen notebook. The desktop tools are familiar and intuitive. And notebook-like tabs let you switch from one page to another in a snap. ProShare software gives you immediate access to all your PC files so you can share any of them instantly.

Only one copy needed to conference.

What about working with colleagues who don't have ProShare Personal Conferencing software? No problem! Modem users can "Jump Start" them by sending a file that enables them to receive a connection. If you're a network user, we'll give you a free Time-Out version of the software to pass along to a colleague.

Intel ProShare software features.

- **On-screen notebook.** The shared notebook provides a familiar yet powerful medium for working together.
- **Mark-up tools and pointers.** Familiar pens, highlighters and text help you communicate with one another.
- **Snapshot and document import.** Take a snapshot of a portion of any Windows document, or import an entire multi-page document, and transfer it into your on-screen notebook.
- **Shared OLE.** Enables you to launch the source application, make changes and instantly update the document in the on-screen notebook.

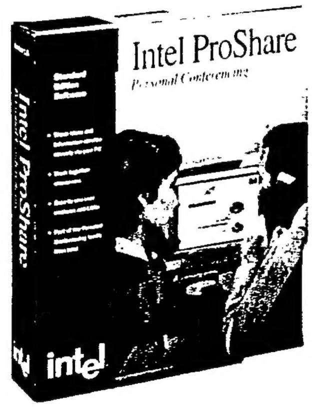
v a r e

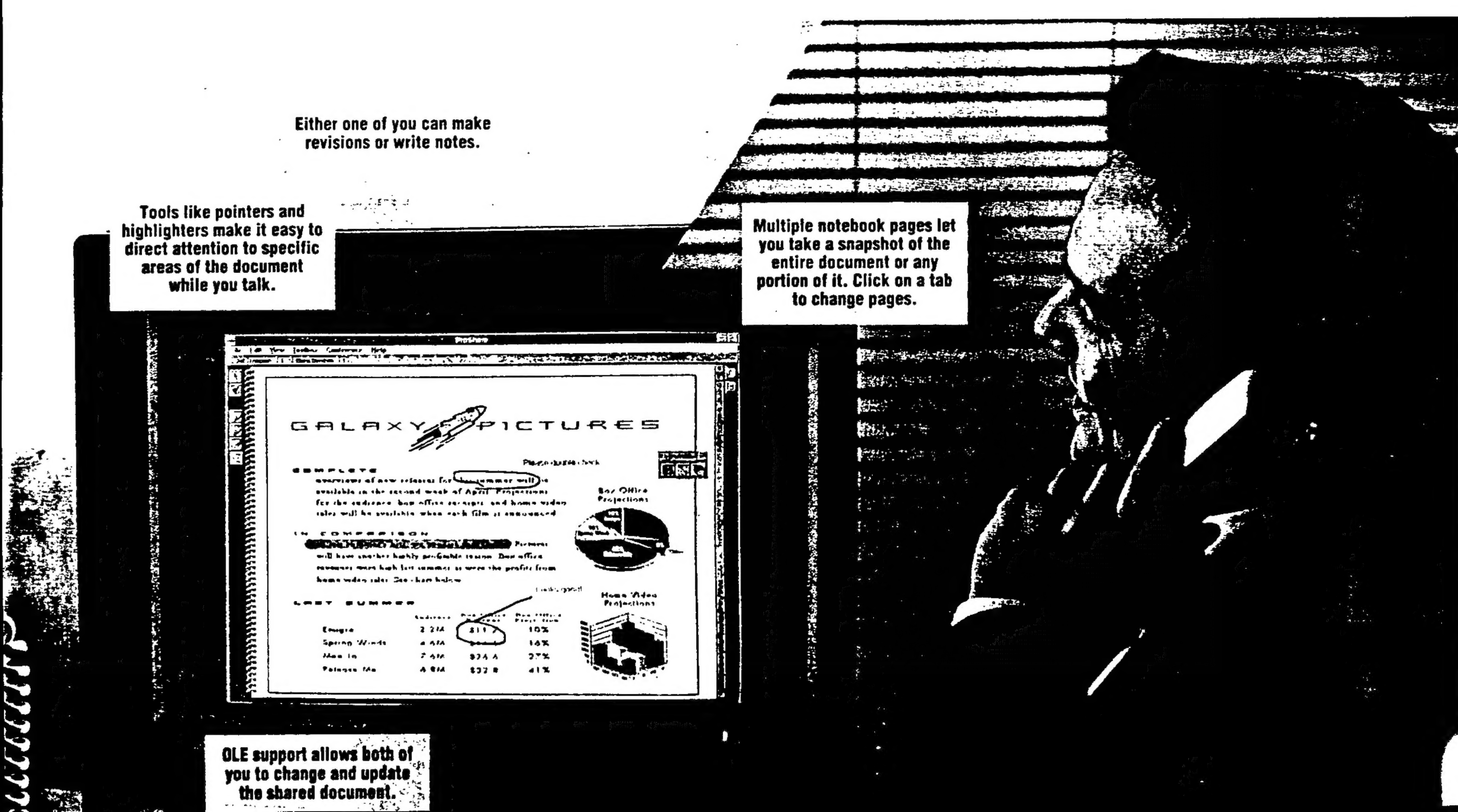
it now.

- **Only one copy needed.** Modem users can download Jump Start, which enables someone to receive a connection. A fully functional Time-Out version is included to pass along to colleagues for use over a LAN.
- **Private space.** Partition your shared notebook to create a private workspace where personal notes are not visible to others.
- **Print and save.** Store your work sessions for reference or later use; or print them in color or black-and-white gray scale.
- **Modem or LAN connection.** Beyond your PC, you only need a modem, phone, and two standard telephone lines (modem use only) to conference. Or conference over the network.
- **Easy to upgrade.** You can add video to document conferencing with a simple upgrade.

Ideas for using Intel's ProShare software.

- A training staff can show remote users how to use applications. They can even take snapshots of screens to leave behind.
- Connect with contractors at a jobsite, such as a construction site or trade show hall, to view and discuss last-minute changes or check progress.
- Research, advertising and PR firms can show clients drafts, creative ideas and schedules and get immediate feedback.
- An architect can go over a house plan with a client in another city and capture the client's feedback and ideas on the drawings.
- An accountant on the fifth floor can view and discuss a spreadsheet with purchasing on the eighth floor.
- A salesperson can present an important multi-page presentation to a client without either leaving their office.





Intel ProShare™ Personal Conferencing Vid

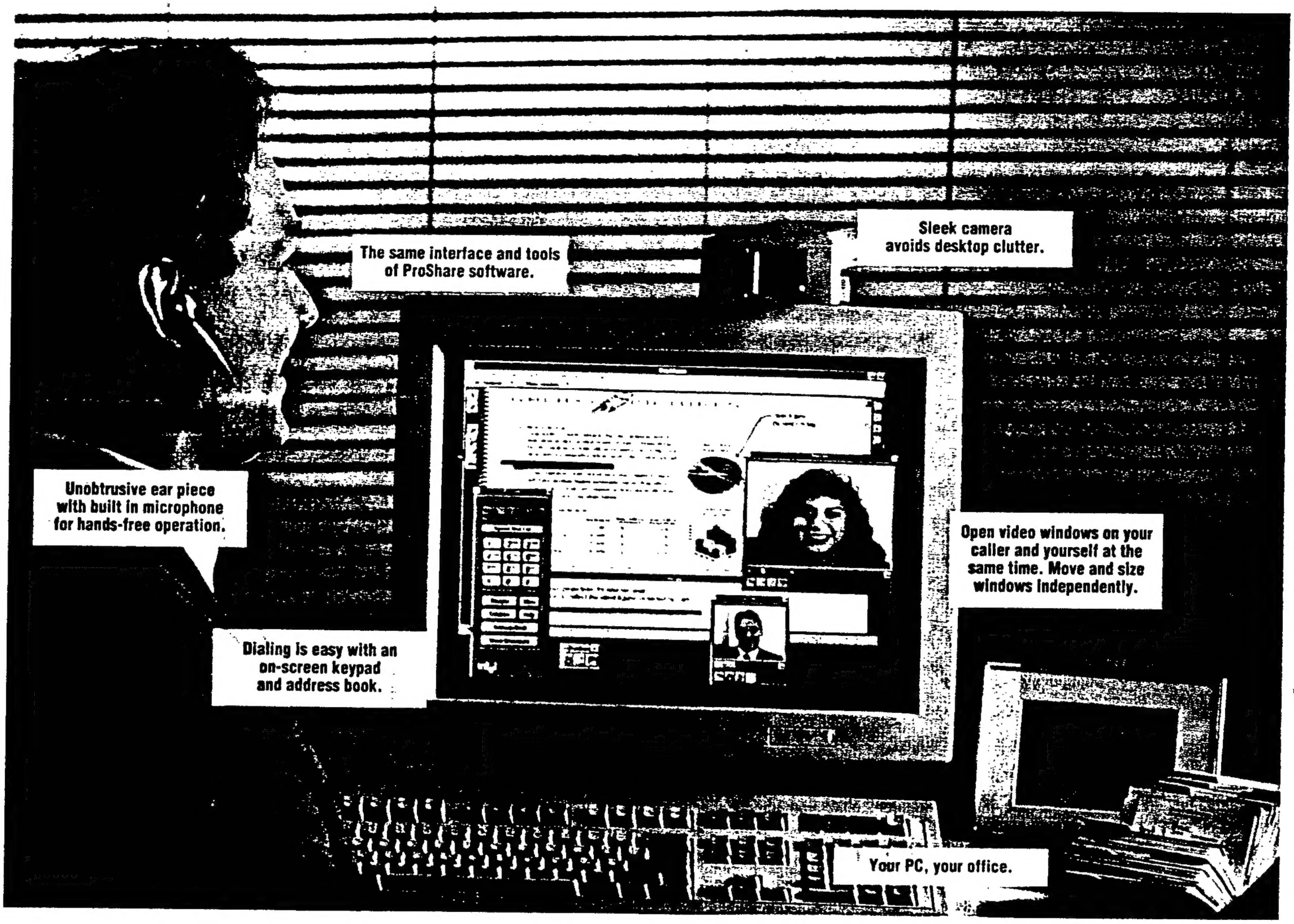
Meet face-to-face and work together rig

If you think ProShare Personal Conferencing software makes communication easier and more efficient, just wait—it gets even better. Say you'd like to be face-to-face as well as document-to-document. You can be with Intel's ProShare Personal Conferencing Video System. All it takes is a simple upgrade to let you use not only your voice but every aspect of your presence to get your message across. Even when your colleague is in another building or another city.

Add a video dimension.

With the Intel ProShare Personal Conferencing Video System installed on your and your colleague's PCs, simply dial their number using the graphical on-screen keypad. It's as easy to operate as the telephone on your desk. Invite your caller into your office via a friendly video window, and simultaneously see how you come across in your own video window. The comfortable ear piece lets you hear everything, doesn't get in the way. Images are as large as a quarter screen thanks to Intel's advanced Indeo® video compression technology.

indeo™
INTEL VIDEO TECHNOLOGY



System

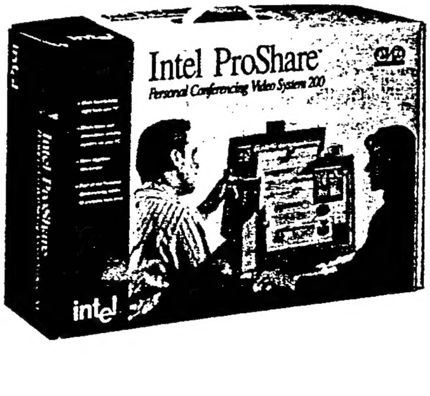
from your PC.

Share a document while you see and hear each other.

You don't just see each other; you can see your work, too. That's because Intel ProShare Video System includes all of the document sharing features contained in ProShare Personal Conferencing software. So you can work on documents together while you meet face-to-face.

Communicate more clearly, make decisions faster.

Intel's ProShare Personal Conferencing Video System revolutionizes routine business communications. Meetings once held by phone are vastly more effective. See your colleague's reactions and express yourself with all your presence to make the maximum impact. No more writing and distributing meeting notes; just save your document conferencing work session. No more time lost in duplicating and shipping objects or images; just show them in the video window and grab a snapshot. It's real-time, immediate communication that gets results now!



Intel ProShare Video System features.

- **Full-featured personal conferencing.** See and hear each other while working together using the shared notebook.
- **Interoperable.** Conference with anyone who has another ProShare product. They're all interoperable. If they don't have video you can still share documents over a modem or LAN.
- **Upgradeable.** If you have a ProShare software product, it's easy to upgrade to video. The user interface is identical.
- **Advanced PC video technology.** With Intel Indeo video, the ProShare Video System supports video window sizes from icon size to one-quarter screen (320 x 240 pixels).
- **Compatible with your existing PC standards.** Capable of recording high quality Indeo video to your hard disk in standard Windows format (.AVI) for later playback (software kit available from Intel). Ready to play and record high quality (16-bit) Windows business audio (.WAV files).
- **Easy to use.** Simple on-screen keypad lets you dial, access built-in address book and share an on-screen notebook.
- **Take high-resolution snapshots.** Take a picture of anything in the video window by clicking on the camera icon.
- **Complete with camera and earphone.** Sleek, miniature camera mounts atop monitor, comfortable ear piece with built-in microphone for hands-free operation.

Ideas for using Intel's ProShare Video System.

- Establish a video connection to your most important customers. You have more presence in the account, a stronger business relationship, and a big competitive advantage.
- Manage product development or project teams. Review schedules, budgets, prototypes, design data, planning details and much more in the shared notebook. Face-to-face communication speeds your time to market by avoiding costly miscommunication.
- Manage more effectively. Dispersed teams can communicate one-on-one, building consensus, setting clear direction and getting better results.

Intel ProShare

Intel ProShare Personal Conferencing software.

System Requirements

- PC with Intel386™ CPU or higher (Intel486™ CPU recommended)
- 4 MB RAM (8 MB recommended)
- 8 MB of available hard disk space
- DOS* 3.3 plus Windows* 3.1 or higher in enhanced mode
- VGA or higher resolution monitor
- For use over LAN or modem
  - Novell NetWare* 3.11 or greater
  - NetBIOS (e.g. Windows for Workgroups 3.X or LANtastic* 5.0)
  - Modem with 9600 bps data rate or higher

Intel ProShare Personal Conferencing Video System.

System Requirements

- PC with Intel486™ 33MHz CPU or higher
- Windows 3.1
- 8 MB RAM, plus 7MB hard disk space (minimum)
- VGA display with 256 colors or higher (no feature connector required)
- Basic-rate 2B+D ISDN phone line with S/T interface. "U" interface requires NT-1 converter

Technical Specifications

- Two full-length 16-bit ISA cards
- Full duplex video and audio conferencing
- Intel Indeo video technology
- 16-bit wave audio
- Jacks for ear piece, microphone, line in and line out
- Standard NTSC composite input
- Supported switches include: AT&T 5ESS (5E6 through 5E9, including NI-1; Northern Telecom DMS-100* (BCS-33 through BCS-36 including NI-1); Siemens EWSD (NI-1)
- 160 x 120, 320 x 240, or icon video window size support
- Up to 24-bit color depth
- Supports all VGA and SVGA resolutions (256 color)
- H.320 option available in late 1994

Do you have any questions?

For more information on Intel ProShare Personal Conferencing products, call one of the numbers below. Or, for instant information by fax, call our FaxBack* information service and use the document number to place your order (touch-tone phones only).

	U.S. AND CANADA	EUROPE	WORLDWIDE
Product Information	800-538-3373	44-793-431155	503-629-7354
FaxBack Information Service	800-525-3019	44-793-432509	503-629-7576

AVAILABLE FAXBACK DOCUMENTS	DOCUMENT NUMBER
ProShare software Modem Compatibility	8100
Understanding ProShare Personal Conferencing	8104
Indeo Video: A Technology Overview	8004
Intel Price List	9000

For ISDN information:

Call Intel's ISDN Connectivity Information Center (ICIC) at 800-538-3373 extension #208.

Printed in USA 0294/75K/IIB/BB